

**IGOR HENRIQUE BELOTI PIZETTA**

**UMA PLATAFORMA PARA SUPORTE À NAVEGAÇÃO  
AUTÔNOMA DE VEÍCULOS AÉREOS DE PÁS ROTATIVAS**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica, na área de concentração Automação.

Orientador: Prof. Dr. Mário Sarcinelli Filho

Coorientador: Prof. Dr. Alexandre Santos Brandão

VITÓRIA, ES - BRASIL

ABRIL DE 2013

Dados Internacionais de Catalogação-na-publicação (CIP)  
(Biblioteca Setorial Tecnológica, Universidade Federal do Espírito Santo, ES, Brasil)

---

Pizetta, Igor Henrique Beloti, 1988-  
P695p      Uma plataforma para suporte à navegação autônoma de  
veículos aéreos de pás rotativas / Igor Henrique Beloti Pizetta. -  
2013.  
128 f. : il.

Orientador: Mário Sarcinelli Filho.

Coorientador: Alexandre Santos Brandão.

Dissertação (Mestrado em Engenharia Elétrica) - Universidade  
Federal do Espírito Santo, Centro Tecnológico.

1. Helicópteros - Características de manobra. 2. Helicópteros -  
Aerodinâmica. 3. Navegação aérea. 4. Navegação de robôs móveis.  
5. Veículo aéreo não tripulado (VANT). 6. Robótica aérea.  
I. Sarcinelli Filho, Mário. II. Brandão, Alexandre Santos  
III. Universidade Federal do Espírito Santo. Centro Tecnológico.  
IV. Título.

CDU: 621.3

---

IGOR HENRIQUE BELOTI PIZETTA

UMA PLATAFORMA PARA SUPORTE À NAVEGAÇÃO  
AUTÔNOMA DE VEÍCULOS AÉREOS DE PÁS ROTATIVAS

Dissertação submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau de Mestre em Engenharia Elétrica - Automação.

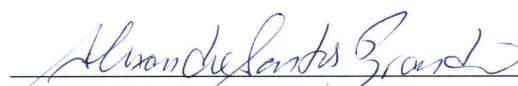
Aprovada em 03 de maio de 2013.

COMISSÃO EXAMINADORA



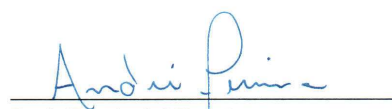
---

Prof. Dr. Mário Sarcinelli Filho  
Universidade Federal do Espírito Santo  
Orientador



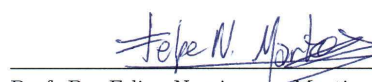
---

Prof. Dr. Alexandre Santos Brandão  
Universidade Federal de Viçosa  
Coorientador



---

Prof. Dr. André Ferreira  
Universidade Federal do Espírito Santo



---

Prof. Dr. Felipe Nascimento Martins  
Instituto Federal do Espírito Santo



*Aos meus pais Madalena e Otacílio  
e a minha avó Tereza.*



# *Agradecimentos*

Aos meus pais, Madalena Maria Beloti Pizetta e Otacílio Pizetta, por sempre me apoiarem nas minhas decisões e ajudar quando precisei, seja financeiramente ou através de conselhos. Também à minha avó Tereza Favalessa e minha tia Maurian Beloti, pelo apoio dado ao longo do mestrado.

Ao meu orientador Prof. Dr. Mário Sarcinelli Filho, que me aceitou como aluno de mestrado e que desde o primeiro dia de trabalho pergunta "*E aí, tá voando?*" ou cantando pelo corredor "*voar voar, subir subir*". Que passava três vezes ao dia no laboratório e falava as três vezes "*E aí?! me conta uma notícia boa*", "*Tá voando?*". O que pode parecer excesso de cobrança, considerava incentivo e, muitas vezes, motivo de descontração no laboratório.

Ao Prof., agora Dr., Alexandre Brandão, por me incentivar a entrar nesse desafio do mestrado e pela confiança em mim depositada, mesmo quando nem eu mesmo sabia o que estava fazendo. Pelos diversos embalos de sábado à noite (ou domingos de tarde) no laboratório. Que a cada vinda a Vitória recarregava as energias e aumentava o desejo pela realização do almejado voo autônomo.

Aos companheiros de  $LAI_1$  Flávio Garcia, Frodo (Fabrício) Baggins (Bortolini), Marinho Frank, Mariana Rampinelli, Rafael Valverde, por nunca negarem um café, seja depois do almoço ou no meio da tarde. Sempre dispostos a uma conversa no meio da tarde quando tudo resolvia não funcionar. Além de nunca negarem uma ajuda quando preciso.

Ao Milton Santos,  $\in LAI_1$ , não só pela convivência em San Juan, mas também pelo sempre interesse no projeto e em ver as coisas funcionando.

Aos amigos do  $LAI_3$ , o trio de colombianos *más que locos*, Camilo Rodriguez, Carlos Cifuentes e Manuel, pelas risadas mesmo quando as mesmas eram exclusivas para colombianos. E também ao Dennis pelas dicas e ajudas com o Linux.

Ao Roberto Salles Melin,  $\in LAI_3$  companheiro de mestrado e muitas vezes de academia. Apesar de que conversava mais do que malhava.

Aos companheiros do laboratório oficial  $LAI_2$ , Valmir, Pâmela, Netaliani, Anibal,

Richard, Jhon<sup>2</sup>, por aguentar os testes e experimentos realizados, com a bravura de muitas vezes ficar sentado em suas cadeiras, a qual eu mesmo não tinha. Também ao Carlos Valadão,  $\in LAI_2$ , por suportar meu ligeiro hiperativismo em San Juan, sempre com sua calma 0 Kelvin.

Ao Lucas Vago, pela inovação que traz ao laboratório e ao projeto, a cada visita. Sempre se prontificando a ajudar no que podia e fazendo diferença no desenvolvimento, seja pelas inovações ou pela ajuda.

A Danielle Mendes Silva, que deixei para o final não por menor importância e sim por não haver palavras para agradecer-lhe. Obrigado pela companhia ao longo desses longos anos, às vezes mais longos que o esperado. Companheira de todas as batalhas, a você só tenho a agradecer.

*A todos vocês muito obrigado!*

*Syer Lyette*



*“Tudo, antes de ser fácil, é difícil.”*

***Thomas Fuller***



# *Resumo*

Esta Dissertação de Mestrado tem por objetivo a criação de uma plataforma para testes de voo autônomo simulados e experimentais usando controladores de alto nível para veículos aéreos em miniatura do tipo helicóptero. Ela utiliza um helimodelo comercial e a instrumentação necessária para a realização de um voo autônomo. Para isto, foram agregados ao veículo sensores inerciais, ultrassônicos, barométricos e GPS, e foi projetada e confeccionada uma placa específica para atuação sobre os servomotores do helicóptero e leitura dos sensores, chamada de *AuRoRa Board*. Ela realiza a leitura de todos os comandos vindos do rádio controle e também recebe os dados vindos do controlador de alto nível para navegação autônoma. A seleção de qual sinal irá atuar no sistema é feita pelo operador através do rádio controle, medida tomada por motivos de segurança (a operação manual desativa a automática quando acionada pelo operador). Neste trabalho também foi utilizado um estabilizador de baixo nível, chamado de *Helicommand*, que tem como objetivo estabilizar o helicóptero em voo pairado, para facilitar o controle da sua navegação e, caso a comunicação seja perdida, manter o veículo estabilizado até se recuperar a comunicação ou o operador assumir o controle do veículo via rádio controle. Também foi criada uma plataforma de alto nível do tipo *Hardware-in-the-Loop*, chamada de Plataforma *AuRoRa*, capaz de realizar simulações e experimentos reais utilizando a placa *AuRoRa Board*. Nela pode-se escolher entre diversos veículos aéreos, dentre eles quadrimotores e os helicópteros miniatura T-REX 450 e T-REX 600, sendo estes dois últimos aqueles com que se trabalhou nesta Dissertação mais diretamente. A plataforma também é capaz de suportar diversos tipos de comunicação, como serial, *Bluetooth* e UDP (*User Datagram Protocol*), além de diferentes tipos de controladores e modelos de veículos. A plataforma apresenta uma característica descentralizada, ou seja, para evitar a sobrecarga de um computador com a exibição *online* dos dados de voo junto com a síntese dos sinais de controle, pode-se delegar certas funções para outros computadores (por exemplo, a exibição *online* dos dados de voo), realizando a comunicação entre os mesmos através de uma rede UDP.



# *Abstract*

This M.Sc. Thesis has the objective of creating a platform to support the test of simulated and real autonomous flights, using high level controllers for miniature aerial vehicles of the class helicopter. Such platform is designed for use with commercial miniature helicopters and embeds the instrumentation necessary for flying autonomously. Aiming at this, an inertial sensor, an ultrasonic sensor, a barometric sensor and a GPS receptor were integrated to the vehicle. Besides, an electronic board specific to actuate over the servomotors of the helicopter, and to get the data delivered by the onboard sensors the aircraft as well, referred to as the *AuRoRa* Board, was designed and manufactured. Such board reads all the commands coming from the radio control and the signals coming from the high level controller responsible for the autonomous navigation of the vehicle. The selection of the signal that will actuate over the servos is performed by the operator through the radio controller, for security reasons (manual operation disables the autonomous operation when activated by the operator). In this work it was also used a low level stabilizer, called Helicommand, whose objective is to estabilize the helicopter hovering in its current position, to make easier the control of its navigation and, in the event of communication loss, to keep the helicopter stabilized until either recovering the communication or a human operator takes the vehicle control via radio. It was also implemented a high level Hardware-in-the-Loop platform, called *AuRoRa* Platform, capable of running simulations and real experiments, these last ones using the *AuRoRa* Board to actuate over the servomotors of the vehicle. In such platform one can choose among various aircrafts, amongst which are the quadrotors and the miniature helicopters T-REX 450 and T-REX 600 (these two last ones were the aircrafts dealt with in this Thesis). The platform is also capable of supporting various communication channels, like serial, Bluetooth and UDP (User Datagram Protocol), besides accepting different controllers and vehicle models. The platform presents a decentralized characteristic: to avoid overloading a single computer with the control signals synthesis and the online exhibition of the flight data one can distribute certain tasks (e. g., online exhibition of flight data) to another computer, using a UDP communication channel.



# *Sumário*

## **Lista de Figuras**

## **Lista de Tabelas**

<b>1</b>	<b>Introdução</b>	<b>25</b>
1.1	Estado da Arte . . . . .	28
1.2	Objetivos . . . . .	30
1.3	Estrutura do Trabalho . . . . .	31
<b>2</b>	<b>Modelagem de Baixo Nível de um Helicóptero Miniatura</b>	<b>35</b>
2.1	Princípio de Funcionamento de um Helimodelo . . . . .	37
2.2	Dinâmica dos Atuadores . . . . .	40
2.3	Dinâmica das Pás Rotativas . . . . .	44
2.4	Processo de Geração de Forças e Torques . . . . .	46
<b>3</b>	<b>Instrumentação de Voo</b>	<b>51</b>
3.1	Unidade Inercial . . . . .	51
3.2	Sensores de Referência Global . . . . .	53
3.3	Microncontrolador PIC24FJ256GB206 . . . . .	55
3.4	Computador Embarcado PicoITX . . . . .	56
3.5	Transmissão de Dados . . . . .	56
3.6	Estabilizador de Baixo Nível . . . . .	58
3.7	Sistema Embarcado <i>AuRoRa Board</i> . . . . .	60

3.8	Placa de Desenvolvimento ARM . . . . .	63
<b>4</b>	<b>A Plataforma <i>AuRoRa</i></b>	<b>65</b>
4.1	Estrutura da Plataforma <i>AuRoRa</i> . . . . .	66
4.2	Estrutura Descentralizada da Plataforma . . . . .	69
4.3	Estrutura Multi-veículos da Plataforma . . . . .	72
<b>5</b>	<b>Exemplos de Utilização da Plataforma</b>	<b>75</b>
5.1	Simulação de Voo com Movimento de Altitude e Guinada . . . . .	75
5.2	Simulação de Voo Tridimensional . . . . .	77
5.3	Leitura dos Dados de Voo . . . . .	77
5.4	Simulação com Vários Veículos . . . . .	82
5.5	Simulação <i>Hardware-in-the-Loop</i> . . . . .	82
5.6	Controle Automático de Altura . . . . .	86
<b>6</b>	<b>Conclusões</b>	<b>89</b>
	<b>Apêndice A – Fusão Sensorial</b>	<b>91</b>
A.1	Quatérnios . . . . .	91
A.2	Operações por Quatérnios . . . . .	92
A.3	Filtragem por Quatérnios . . . . .	94
A.4	Orientação a Partir de Observação de Vetores . . . . .	95
A.4.1	Algoritmo de Fusão . . . . .	98
A.4.2	Compensação da Distorção Magnética . . . . .	100
A.4.3	Compensação do <i>Drift</i> do Giroscópio . . . . .	100
	<b>Apêndice B – AuRoRa Board</b>	<b>103</b>
B.1	Códigos da Plataforma <i>AuRoRa</i> Comuns aos Helimodelos TREX-450 e TREX-600 . . . . .	103



B.1.1	main.c . . . . .	103
B.1.2	Init_PWM.c . . . . .	105
B.1.3	Init_ADC.c . . . . .	107
B.1.4	Init_UART2.c . . . . .	107
B.1.5	Init_IC . . . . .	109
B.1.6	Init_UART1 . . . . .	114
B.1.7	TimerInterrrupts.c . . . . .	116
B.2	Códigos Diferenciados por Veículo . . . . .	117
B.2.1	Init_IO_450 . . . . .	117
B.2.2	Init_IO600 . . . . .	118
B.3	Circuito da Placa <i>AuRoRa Board</i> . . . . .	119
B.4	Camadas Físicas da Placa <i>AuRoRa Board</i> . . . . .	120



# *Lista de Figuras*

1	Exemplo da utilização de veículos aéreos não tripulados (VANTs). . . . .	25
2	Taxonomia dos VANTs atualmente utilizados. . . . .	26
3	Representação da mobilidade dos veículos aéreos de asas fixas e de pás rotativas. . . . .	26
4	Desenvolvimento de VANTs por país[1]. . . . .	27
5	Representação em blocos do modelo dinâmico de um helicóptero. . . . .	36
6	Sistemas de referência utilizados. . . . .	38
7	A representação de um helimodelo com seus graus de liberdade. . . . .	38
8	Passos cíclicos. . . . .	39
9	Passos coletivos. . . . .	40
10	Servomotor utilizado em helimodelismo. . . . .	41
11	Sinal PWM usado no controle do servomotor. . . . .	41
12	Rádios utilizados nesta Dissertação (fabricante: SPEKTRUM). . . . .	42
13	Separação dos canais no sinal PPM. . . . .	42
14	<i>Swashplate</i> (bailarina ou prato oscilante), responsável por converter as entradas de controle em movimento das pás do rotor principal e das barras estabilizadoras. . . . .	43
15	Diagrama do funcionamento do comando de coletivo. . . . .	43
16	Diagrama do funcionamento do comando de guinada. . . . .	43
17	Diagrama do funcionamento do comando de arfagem. . . . .	44
18	Diagrama do funcionamento do comando de rolagem. . . . .	44
19	Ângulo de ataque $\Theta(\Psi)$ das pás do helicóptero. . . . .	45

20	Posição angular das pás ( $\Psi$ ) ao redor do eixo do rotor principal. . . . .	46
21	Ângulos que determinam a orientação da propulsão. . . . .	46
22	Definição dos braços para as forças aplicadas ao helicóptero, responsáveis pelos torques aplicados à aeronave. . . . .	48
23	9DOF Razor IMU - <i>Sparkfun</i> . . . . .	52
24	BMP085 - <i>Sparkfun</i> . . . . .	54
25	EZ4 MaxBotix - <i>Sparkfun</i> . . . . .	55
26	Microcontrolador PIC24FJ256GB206 - <i>Microchip</i> . . . . .	55
27	Pico ITX - <i>EPIA</i> . . . . .	56
28	Representação do pacote de dados do protocolo UDP. . . . .	57
29	XBee - <i>Maxstream</i> . . . . .	58
30	Módulo Bluetooth JY-MCU. . . . .	58
31	Helicommand 3D - <i>Captron</i> . . . . .	59
32	Conexão do módulo Helicommand ao sistema de recepção de rádio do helimodelo. . . . .	60
33	Imagem da placa <i>AuRoRa Board</i> (Os conectores foram omitidos para me- lhor visualização das placas). . . . .	61
34	Esquemático da placa <i>AuRoRa Board</i> . . . . .	61
35	Ilustração da conexão da <i>AuRoRa Board</i> ao sistema de recepção de sinais. . . . .	63
36	Placa de desenvolvimento ARM STM32F3DISCOVERY da ST Instruments. . . . .	64
37	Diagrama de blocos do funcionamento interno da Plataforma <i>AuRoRa</i> . . . . .	67
38	Estrutura da plataforma experimental com distribuição de tarefas. . . . .	70
39	Diagrama de blocos do funcionamento interno da plataforma desenvolvida. . . . .	71
40	Duração das iterações da plataforma <i>AuRoRa</i> . . . . .	72
41	Apresentação de uma simulação para comparação de dois modelos. . . . .	73
42	Diagrama de blocos do funcionamento interno da plataforma desenvolvida para multi-veículos. . . . .	74

43	Simulação: Controle de altitude e guinada utilizando um helicóptero miniatura. . . . .	76
44	Simulação: Seguimento de trajetória em forma de 8 sem restrição de movimento, usando helicóptero miniatura. . . . .	78
45	Leituras dos dados provindos do rádio controle do helimodelo T-REX 450. . . . .	79
46	Valores de tensão da bateria durante um experimento. . . . .	80
47	Valores do coletivo, velocidade e altura do helimodelo T-REX 600. . . . .	81
48	Experimento com T-REX 600. . . . .	81
49	Simulação de uma tarefa que envolve a coordenação entre dois VANTs. . . . .	83
50	Simulação HIL para teste da unidade inercial. . . . .	84
51	Tempo de execução das iterações com a comunicação <i>Bluetooth</i> . . . . .	85
52	Controle autônomo da altura. . . . .	86
53	Controle autônomo da altura. . . . .	87
54	Controle autônomo da altura. . . . .	87
55	Orientação dos eixos coordenados de de B para A utiliznado quartérnios. . . . .	92
56	Diagrama do circuito da placa <i>AuRoRa Board</i> . . . . .	119
57	Parte superior da placa <i>AuRoRa Board</i> . . . . .	120
58	Parte inferior da placa <i>AuRoRa Board</i> . . . . .	120



## *Lista de Tabelas*

- 1    Comparação entre os VANTs de asas fixas e de pás rotativas, tendo em conta algumas características básicas de voo [2, 3]. **Nota:** 1 é ruim e 3 é excelente. . . . . 27





# 1 Introdução

*“Pesquisa é o que eu faço quando não sei o que estou fazendo.”*

*(Wernher von Braun)*

O desenvolvimento de pesquisas com veículos aéreos não tripulados (VANT, do inglês, UAV - *Unmanned Aerial Vehicles*), tem crescido bastante, tanto no meio acadêmico quanto no meio industrial. Isso se deve às muitas possibilidades de aplicação de VANTs, dentre as quais se pode citar segurança pública (supervisão do espaço aéreo e de tráfego urbano), gerenciamento de riscos naturais (como vulcões, por exemplo), gerenciamento ambiental (medida de poluição do ar, supervisão de florestas, verificação de derramamento de óleo), intervenção em ambientes hostis, manutenção de infraestruturas (linhas de transmissão, dutos de líquido ou gás), agricultura de precisão, cinematografia e exploração (citando apenas aplicações civis de tais aeronaves [4, 5, 6, 7, 8, 9]). A Figura 1 mostra alguns desses veículos em ação.

Os VANTs podem ser classificados em dois grandes grupos: mais leves que o ar e mais pesados que o ar, como se pode ver na Figura 2. Neste trabalho, serão considerados apenas VANTs do segundo grupo, os veículos mais pesados que o ar. Nele encontram-se os VANTs de asas fixas (aviões e planadores) e os de pás rotativas (helicóptero convencional e quadrimotores, por exemplo). Uma diferença marcante entre estas duas famílias de



(a) Inspeção de linhas de transmissão.



(b) Fotografia aérea.



(c) Utilização em missões militares.

Figura 1: Exemplo da utilização de veículos aéreos não tripulados (VANTs).

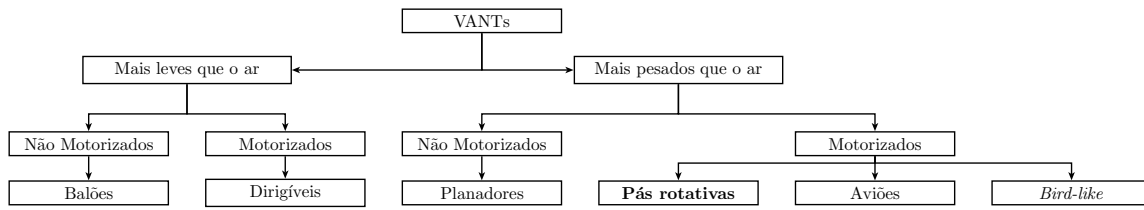


Figura 2: Taxonomia dos VANTs atualmente utilizados.

aeronaves é a mobilidade tridimensional: veículos de pás rotativas possuem a capacidade de decolar, planar e aterrissar sem a necessidade de movimentação longitudinal, como os aviões, e por isto podem ser utilizados em ambientes de pequenas dimensões, como laboratórios, galpões e escritórios, dentre outros [10].

Essa mobilidade tridimensional diferenciada dos veículos aéreos de pás rotativas os tornam, do ponto de vista físico, umas das máquinas voadoras mais complexas, com capacidade de decolar e aterrissar verticalmente, realizar tarefas de voo pairado alterando sua guinada, mover-se longitudinalmente e/ou lateralmente enquanto mantêm a mesma cota vertical, além de poderem alterar completamente a sua direção de voo, ou mesmo se deterem de forma repentina [11, 12]. A Figura 3 ilustra a mobilidade tridimensional das aeronaves de asas fixas e de pás rotativas. Destaque-se que para o caso da Figura 3(a) é sempre necessário manter uma velocidade  $v$ , no sentido longitudinal, para a realização de seus possíveis movimentos.

Como consequência dessa manobrabilidade, tem-se a complexidade de controlar tais veículos, por serem sistemas inerentemente instáveis, multi-variáveis, com dinâmica complexa e altamente acoplada. Isso torna um desafio o desenvolvimento de sistemas aéreos não tripulados (SANT, do inglês UAS - *Unmanned Aerial System*); porém, as aplicações destes veículos têm motivado diversas pesquisas ao redor do mundo, com as dificuldades

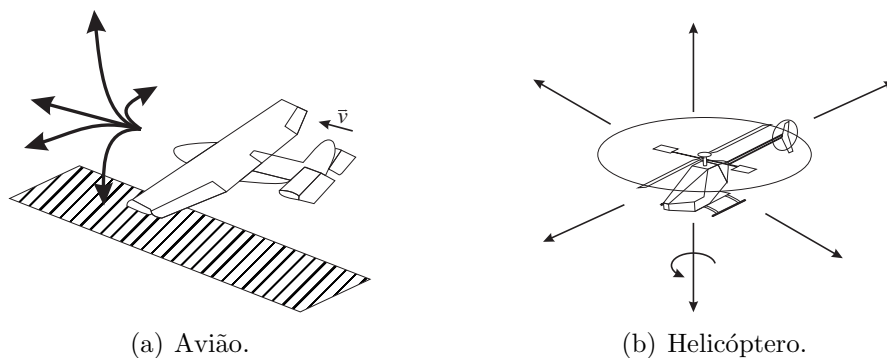


Figura 3: Representação da mobilidade dos veículos aéreos de asas fixas e de pás rotativas.

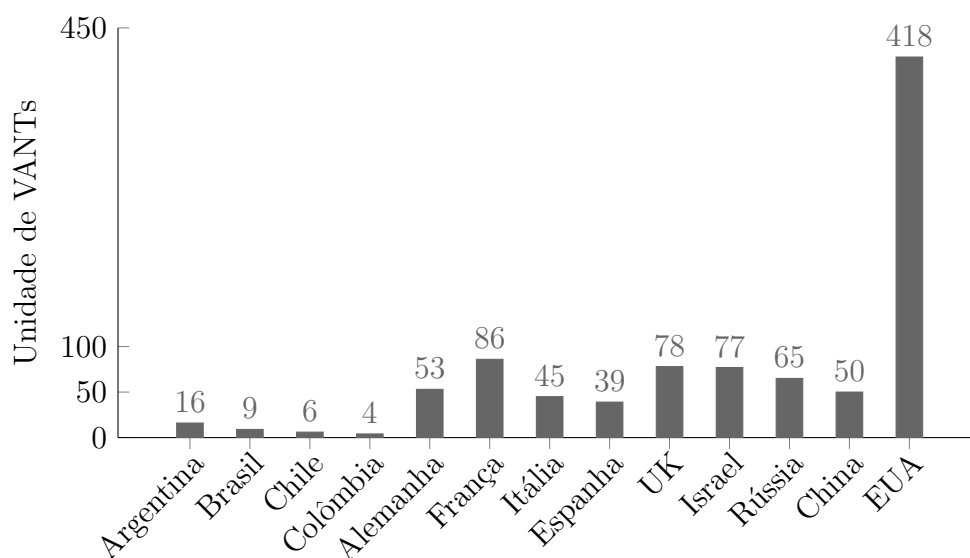


Figura 4: Desenvolvimento de VANTs por país[1].

acima mencionadas sendo encaradas como desafios. Em [1] é apresentada uma perspectiva de tais sistemas no cenário mundial, enquanto a Figura 4 apresenta o número de projetos desenvolvidos, ou em desenvolvimento, de alguns países ao redor do mundo. São notáveis a liderança dos Estados Unidos da América no número de pesquisas e a pouca participação dos países da América do Sul. Para este relatório, foram analisados os veículos classificados como Mini, o que significa veículos com massa inferior a  $30kg$ .

A Tabela 1 apresenta uma análise comparativa entre VANTs de asas fixas e rotativas, analisando oito características básicas de voo. Apesar de um alto consumo de energia, os veículos de asas rotativas apresentam vantagem significativa quando comparados aos veículos de asas fixas. Em particular, eles atendem melhor às condições de trabalho desta Dissertação, uma vez que permitem testes em ambientes *indoor* (laboratórios e ginásios),

Tabela 1: Comparação entre os VANTs de asas fixas e de pás rotativas, tendo em conta algumas características básicas de voo [2, 3]. **Nota:** 1 é ruim e 3 é excelente.

Característica	Asas fixas	Asas rotativas
Consumo de Energia	2	1
Manobrabilidade	2	3
Voo estacionário	1	3
Voo a baixas velocidades	1	3
Voo a altas velocidades	3	2
Miniaturização	2	3
Decolagem vertical	1	3
Utilização <i>indoor</i>	1	3
Total	12	21

não necessitando, assim, grandes espaços para experimentos.

Um problema associado a testes com VANTs é o perigo envolvido, não somente para o equipamento como também para os pesquisadores. Por este motivo, é sábio ter resultados simulados do sistema desenvolvido antes de realizar testes experimentais. Desta forma, diversos pesquisadores têm implementado suas próprias plataformas de desenvolvimento, ou seja, sistemas integrados capazes de realizar simulações para confirmar se o sistema desenvolvido está apto para testes experimentais [13, 14, 15, 16, 17, 18, 19, 20].

Neste contexto, a proposta desta Dissertação é a criação de um *hardware/software* específico para simulação e teste de controladores de voo desenvolvidos para veículos aéreos de pás rotativas, além de fornecer todos os dados sensoriais e os sinais de controle, permitindo, assim, analisar o andamento de uma missão de voo. O objetivo é criar uma plataforma versátil, capaz de realizar simulações dos tipos *software-in-the-loop* e *hardware-in-the-loop* [18, 21, 17], possibilitando diferentes configurações para testes e experimentos, agregando, também, um sistema para visualização *online* das variáveis de controle e dos dados de postura da aeronave.

## 1.1 Estado da Arte

A alta aplicabilidade de VANTS e sua superioridade em comparação com os veículos terrestres não tripulados - VTNT, em certas aplicações, juntamente com avanços chave em sistemas sensoriais e embarcados, como diminuição de tamanho e de consumo de energia e aumento da capacidade de processamento, propiciaram um grande crescimento do esforço dedicado à pesquisa nesta área [13]. Porém, o controle de tais aeronaves e a criação de SANTS é o maior dos desafios, tendo motivado diversas pesquisas nas áreas de controle linear, não linear e inteligente, além do desenvolvimento de sistemas sensoriais cada vez mais precisos e sistemas embarcados cada vez mais leves e com maior capacidade computacional, buscando embarcar os controladores desenvolvidos.

Tradicionalmente, o controle autônomo das aeronaves é feito através de laços de controle internos e externos, que são responsáveis, respectivamente, pela estabilização da dinâmica da aeronave e por sua navegação [22]. Porém, mesmo garantindo a estabilidade de cada um desses laços de controle de forma independente, a estabilidade dos dois controladores de forma conjunta não é garantida, devido ao grande acoplamento dinâmico característico de VANTs. Existe, porém, uma linha de pesquisa que trata da solução integrada destes dois problemas, cinemático e dinâmico (navegação e estabilização), que

se baseia em controladores capazes de estabilizar a aeronave e guiar sua navegação de forma simultânea, utilizando técnicas não lineares [23, 24].

Com o intuito da navegação autônoma de um helicóptero miniatura, encontra-se na literatura uma diversidade de estratégias de controle. Em [25] utiliza-se de um sistema baseado em redes neurais artificiais, algoritmos genéticos, controladores clássicos PID e lógica Fuzzy, que são aplicadas, respectivamente, no treinamento e modelagem das características dinâmicas de um helicóptero, na busca e controle dos ângulos de pás do rotor principal e de cauda, além do controle de velocidade do rotor principal. Em [26] é apresentada uma estratégia de seguimento de trajetória de um helicóptero miniatura com um modelo simplificado, denominado *Planar Vertical Takeoff and Landing* - PVTOL - utilizando sistemas Liovillian. Em [27] um sistema de controle de postura de um PVTOL é proposto, utilizando linearização do modelo dinâmico simplificado. A análise de estabilidade é realizada através da teoria de Lyapunov aplicada a sistemas lineares. Em [28] e [29], o modelo dinâmico linearizado da aeronave autônoma é utilizado em tarefas de voo estacionário usando, respectivamente, um controlador PD e um controlador de seguimento de trajetória baseado em técnicas de controle ótimo. Em [30] e [31], um controlador adaptativo neural e um conjunto de controladores PID em cascata são aplicados, respectivamente, para o controle de postura de um VANT. Em [32] é apresentada a proposta de um controlador robusto de seguimento de trajetória considerando as incertezas paramétricas de um VANT em tarefas de movimento longitudinal, lateral, vertical e de guinada. Em [33] um controlador robusto com observadores de estado foi aplicado em um modelo não linear de um helicóptero miniatura sujeito a rajadas de vento vertical. Em [34] um controlador PID com ganho *feedforward* foi utilizado para a compensação de distúrbios no controle de guinada, distúrbios estes inerentes à variação do passo coletivo da hélice do rotor principal durante o controle de altitude.

Por outro lado, diversos trabalhos criam suas próprias plataformas de desenvolvimento, ou seja, dispõem de arquiteturas específicas tanto de *hardware* quando de *software*. Em [21] é desenvolvida uma plataforma *hardware-in-the-loop* para verificação do desempenho dos controladores antes de testes reais. Para tal, o sistema dispõe de dois módulos embarcados, um para atuação e outro para controle, além de uma estação em terra, que fornece os dados de navegação para o módulo embarcado. A estrutura desenvolvida permite a simulação e teste de partes do *hardware* com diferentes níveis de complexidade, tendo o computador em terra o único objetivo de observação e monitoramento, enquanto o de controle fornece as informações, sejam estas simuladas ou reais.

Em [17] é apresentada a configuração dos testes *hardware-in-the-loop* realizados para a determinação da eficácia do sistema de atuadores e a estrutura do sistema desenvolvido, que possui, inclusive, simulações 3D realizadas na estação em terra. Para a comprovação do sistema desenvolvido, em [16] foram realizados testes simulados com a utilização do *software* X-PLANE, que é um simulador de voo que leva em consideração características do modelo físico do veículo e influências externas sobre o mesmo.

Em [13] é apresentado o sistema embarcado para captura de dados de voo de um helicóptero miniatura, onde *software* e *hardware* foram desenvolvidos, este último com a utilização do computador PC-104. Já em [14] é apresentada a estrutura do *software* da plataforma do VANT Berkeley, projeto desenvolvido pelos autores, utilizando como sistema embarcado também um PC-104, com a utilização de uma rede *ad-hoc* para intercomunicação entre os sistemas embarcado e em terra.

Em [15] é apresentada a estrutura da plataforma para um VANT de asas fixas, que dispõe de uma rede *bluetooth* embarcada, onde diversos módulos entregam informações ao principal e este realiza o controle, que pode ser embarcado, realizado por um computador em terra ou até mesmo acionando um modo de emergência e controlando o VANT pelo rádio controle.

## 1.2 Objetivos

A presente Dissertação de Mestrado visa a criação de um sistema embarcado para atuação em baixo nível dos servo motores de um helimodelo tipo helicóptero, assim como a comunicação entre sistemas de baixo e alto níveis e o fornecimento de dados sensoriais necessários para a realização de um voo autônomo, versátil a ponto de ser possível sua utilização com diferentes tipos de helicópteros miniatura. Para isso, é necessário criar um *hardware* específico para tais tarefas, sendo que este deve ser leve e com baixo consumo de energia. -Pretende-se criar uma plataforma de alto nível para gerenciamento dos controladores e modelos, a qual deve ser robusta e contemplar experimentos e simulações, *software-in-the-loop* e *hardware-in-the-loop*. Desta forma, pode-se comprovar o funcionamento das diversas partes desenvolvidas, reduzindo, assim, riscos inerentes ao mal funcionamento de módulos durante experimentos, visando a segurança do equipamento e das pessoas envolvidas. Além disso, deve-se permitir o controle por parte de um usuário, caso necessário. Ou seja, em situações de emergência, a plataforma deve propiciar a opção de desabilitar o controle autônomo e acionar um controle manual, para levar o

equipamento com segurança a um local determinado.

Desta forma, os objetivos específicos são:

- a. estudo da estrutura do helimodelo;
- b. estudo, instalação e configuração do estabilizador de baixo nível Helicommand;
- c. estudo, instalação e configuração do sistema embarcado PicoITX;
- d. projeto de um sistema microcontrolado capaz de capturar os dados sensoriais e se comunicar com o sistema embarcado e com o sistema em terra;
- e. análise e filtragem dos dados sensoriais;
- f. modelagem dos atuadores;
- g. adaptação do sistema para a plataforma desenvolvida.

## 1.3 Estrutura do Trabalho

A organização desta Dissertação segue a estrutura abaixo:

### Capítulo 1: Introdução

Este capítulo apresenta uma breve introdução sobre o tema abordado na Dissertação. Ele introduz, primeiramente, as possíveis utilidades práticas de robôs aéreos, seguindo com a apresentação de uma taxonomia dos mesmos e uma comparação de desempenho entre veículos aéreos de asas fixas (aviões, por exemplo) e de pás rotativas (helicóptero, por exemplo), considerando alguns aspectos relacionados ao voo de tais veículos. Após isto, são mencionados alguns trabalhos relativos ao desenvolvimento de controladores para estes veículos, assim como alguns trabalhos em que foram criadas plataformas de teste para os mesmos, foco principal deste trabalho.

### Capítulo 2: Modelagem de Baixo Nível de um Helicóptero Miniatura

Este capítulo descreve o modelo de baixo nível do helimodelo, apresentando inicialmente os sistemas de referência utilizados e os comandos disponíveis para o controle do mesmo, assim como eles são gerados e como atuam no helimodelo real. Mais adiante é apresentado o modelo de baixo nível, até o subsistema correspondente à

geração das forças e torques que atuam na aeronave, considerada aqui como um corpo rígido. Espera-se que ao final do capítulo o leitor tenha compreendido como o modelo matemático representa o sistema físico, e quais são os comandos disponíveis para se controlar o helimodelo.

### Capítulo 3: Instrumentação de Voo

Neste capítulo são apresentados os sensores utilizados para a instrumentação da aeronave, assim como características físicas de cada um deles. Também é mostrada a forma como os mesmos se intercomunicam, assim como formas de transmissão de dados. Também são mostrados os componentes chave do sensoriamento e as unidades de processamento. É apresentado um estabilizador de baixo nível, denominado Helicommand, seu funcionamento e os objetivos da introdução do mesmo no sistema. Também são apresentadas as características do sistema de processamento desenvolvido, juntamente com seu objetivo e como fica a resultante da interligação física no helimodelo.

### Capítulo 4: Plataforma *AuRoRa*

Neste capítulo é apresentada a plataforma em que se realizam as missões de voo, assim como sua estrutura, que é o foco principal desta Dissertação. É mostrado tanto a parte centralizada, onde todos os cálculos são realizados em um computador, quanto sua parte descentralizada, onde é possível distribuir a parte de geração de gráficos como o modelo do VANT. Por fim é apresentada a possibilidade de realização de testes com múltiplos veículos, lembrando que todas as funcionalidades para simulações podem ser aplicadas para experimentos.

### Capítulo 5: Exemplos de Utilização da Plataforma

Neste capítulo são apresentados vários exemplos de simulação estritamente via *software*, de aplicações do tipo *Hardware-in-the-loop* e experimentos reais.

### Capítulo 6: Conclusões

Neste capítulo são apresentadas algumas conclusões a respeito da plataforma desenvolvida.

Ao final desses capítulos são introduzidos três apêndices, dois deles, A e B, contemplando partes específicas do trabalho que não se enquadram no escopo geral desta Dissertação, porém são pertinentes ao seu desenvolvimento. O Apêndice A apresenta a fusão sensorial utilizada para a obtenção dos ângulos de arfagem, rolagem e guinada



para realimentação do controlador. Já o Apêndice B traz detalhes sobre o código do microcontrolador correspondente ao *software* da *AuRoRa Board*, assim como o circuito correspondente para instalação a bordo do helimodelo, responsável pela interface de comunicação de dados helimodelo-base em terra e pela atuação sobre os servomotores a partir do sinal gerado pelo controlador, além de permitir a navegação por controle remoto via rádio.



## 2 *Modelagem de Baixo Nível de um Helicóptero Miniatura*

*“O trabalho do indivíduo continua a ser a faísca que move a humanidade para a frente.”*

*(Igor Sikorsky<sup>1</sup>)*

Para a realização de um voo autônomo, é necessário a junção entre a modelagem deste veículo e um controlador. Apesar de não ser o objetivo desta dissertação, a noção do modelo do helicóptero é necessária para o entendimento do funcionamento do sistema como um todo. Para mais detalhes sobre tal modelagem, recomenda-se ao leitor consultar [35], [36] e [24].

Para a modelagem matemática, dispõe-se de duas abordagens presentes na literatura, uma baseada nas equações físicas do sistema, que utiliza equações de movimento da mecânica, e outra baseada em técnicas de identificação de sistemas, que estima o modelo dinâmico com base nos dados de excitação e resposta de um sistema. Tais técnicas, porém, nem sempre são excludentes: pode ser necessário utilizar uma para simplificação ou complemento da outra, por exemplo [34, 37].

Em relação à identificação de sistemas pode-se citar diversos trabalhos, como por exemplo [25], no qual foram utilizados dados de voo, juntamente com teoria aerodinâmica, para o treinamento de uma rede neural para guiar o veículo de forma autônoma. Já em [38] foram utilizadas técnicas de modelagem ARM, ARMAX e *Output-Error* para o projeto de um controlador robusto  $H_\infty$ .

Em se tratando de modelagem matemática, pelo fato do VANT estar sujeito a forças e torques externos, é possível sua representação como um corpo rígido que se move no espaço tridimensional. Duas abordagens, então, podem ser adotadas: equações de Newton-Euler [39, 40, 41, 42, 29, 34] e equações de Euler-Lagrange [11, 12, 43]. Ambas abordagens

---

<sup>1</sup>Igor Sikorsky foi pioneiro da aviação e inventor do helicóptero.

levam à representação do modelo dinâmico de corpo rígido. A principal diferença entre elas está em seu sistema de referência. Enquanto a abordagem via equações de Newton-Euler utiliza o sistema de referências no centro de massa do veículo, a modelagem via equações de Euler-Lagrange adota o sistema de referência coincidente com o referencial inercial.

A representação dos modelos no sistema Newton-Euler é

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}, \mathbf{u}, t), \quad (2.1)$$

onde  $\mathbf{x}$  representa as variáveis a serem controladas (postura da aeronave),  $\mathbf{u}$  é o vetor que contém os sinais de controle e  $f(\cdot)$  é uma função, normalmente não linear e dependente do tempo  $t$ , que relaciona o vetor de estados com as entradas de controle. Por outro lado, os modelos baseados nas equações de Euler-Lagrange são representados na forma

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}(\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} + \mathbf{D}, \quad (2.2)$$

onde  $\mathbf{q}$  é o vetor contendo a postura da aeronave,  $\mathbf{M}$  é a matriz de inércia,  $\mathbf{C}$  é a matriz de forças centrípetas e de Coriolis,  $\mathbf{F}$  é o vetor de fricção,  $\mathbf{G}$  é o vetor de força gravitacional,  $\boldsymbol{\tau}$  é o vetor contendo os sinais de controle aplicados ao veículo e  $\mathbf{D}$  é um vetor de distúrbios.

O modelo completo de um helicóptero pode ser considerado, segundo [39, 37, 44], como a combinação de quatro sistemas conectados em cascata, os quais contemplam a modelagem dos atuadores da aeronave e a modelagem do corpo rígido, conforme mostrado na Figura 5.

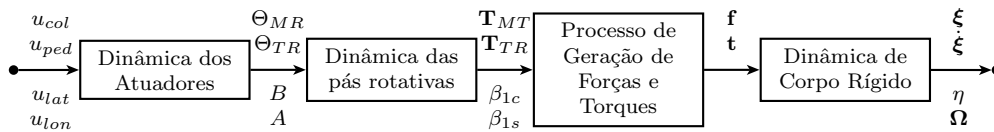


Figura 5: Representação em blocos do modelo dinâmico de um helicóptero.

O bloco intitulado Dinâmica dos Atuadores é onde os comandos enviados pelo rádio controle, no caso de voo controlado remotamente por um operador humano, ou por um sistema computacional a bordo da aeronave ou em terra, no caso de voo autônomo, são recebidos e transformados em ações de comando pelos servo motores. O segundo bloco, Dinâmica das Pás Rotativas, relaciona os parâmetros aerodinâmicos juntamente com a geração de propulsão associada aos motores de cauda e principal. O Processo de Geração de Forças e Torques decompõe esses vetores de propulsão, conforme geometria e a

orientação atual da aeronave, e os transforma em forças e torques aplicados no corpo rígido do veículo. Por fim, a Dinâmica de Corpo Rígido define o deslocamento e a orientação da aeronave no espaço tridimensional, como função das forças e torques aplicados.

Então, os três primeiros blocos mostrados na Figura 5 são responsáveis pela recepção dos sinais de controle e pela geração das forças e torques que irão efetivamente atuar na aeronave. Tais blocos compõem o que será aqui considerado Modelo de Baixo Nível. Por outro lado, os bloco subsequente recebe sinais de controle abstratos (as forças e torques que atuam no corpo da aeronave), os quais irão atuar no modelo de corpo rígido do veículo para sua movimentação no espaço 3-D. Tal bloco, então, forma o Modelo de Alto Nível. Nesta Dissertação, porém, somente o modelo de baixo nível será considerado,

As relações entre os sinais de controle dos servos e as forças e torques, segundo [45], podem ser aproximadas por uma função linear, sempre que se tratar de veículos com massa inferior a 20kg. Em adição os autores ainda comentam que esta aproximação mostrou-se verdadeira experimentalmente. Estas aproximações foram confirmadas também em trabalhos prévios de nosso grupo (ver [46]), onde um controle de altitude e guinada foi implementado em um helicóptero elétrico miniatura TREX 450. Nesta Dissertação também se faz uma análise destas aproximações, agora considerando um helimodelo em escala um pouco maior, o TREX 600, do mesmo fabricante.

De forma geral, esta Dissertação irá apresentar brevemente o modelo de baixo nível apresentado na Figura 5, pois a plataforma desenvolvida inclui um sistema de atuação dos servo motores do helimodelo, e os blocos que compõem o modelo de baixo nível são afetados diretamente pelo sistema desenvolvido. Um detalhamento dos modelos de alto e baixo nível, assim como o projeto de controladores baseados no modelo de alto nível, pode ser encontrado em [47].

## 2.1 Princípio de Funcionamento de um Helimodelo

Para descrever a dinâmica do helicóptero serão definidas notações e referências para melhor entendimento. Desta forma, representa-se pelo subscrito  $e$  o sistema de referência terrestre (inercial), pelo subscrito  $h$  o sistema de referência do helicóptero, centrado em seu centro de massa, e pelo subscrito  $s$  o sistema espacial, que é o sistema  $h$  rotacionado para ter a mesma orientação que  $e$ . As representações destes sistemas podem ser vistas na Figura 6.

As variáveis que representam a mobilidade tridimensional podem ser vistas na Figura

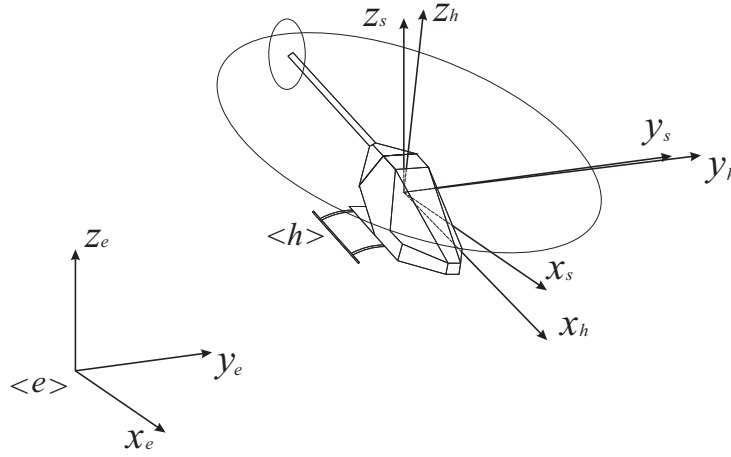


Figura 6: Sistemas de referência utilizados.

7, onde são apresentadas as variáveis de posição  $\xi_{he}$ , onde o subscrito  $he$  indica o sistema de referência do helicóptero relativo ao referencial terrestre, e de orientação  $\eta_{he}$ , relativa a seu próprio sistema de referência. Assim sendo, tem-se que  $\xi_{he} = [x \ y \ z] \in \mathbb{R}^3$  representa os deslocamentos longitudinal, lateral e normal, enquanto que  $\eta_{he} = [\phi \ \theta \ \psi] \in \mathbb{R}^3$  corresponde aos ângulos de rolagem, arfagem e guinada (ver Figura 7).

A fim de determinar as velocidades, basta a derivada das coordenadas de posição, resultando em  $\dot{\xi}_{se} = [\dot{x} \ \dot{y} \ \dot{z}] \in \mathbb{R}^3$ . Derivando novamente obtém-se a aceleração  $\ddot{\xi}_{se} = [\ddot{x} \ \ddot{y} \ \ddot{z}] \in \mathbb{R}^3$ . Note-se que desta vez a referência em questão é a espacial,  $s$ , em relação à terrestre  $e$ . Analogamente faz-se o mesmo para os ângulos, tendo, assim, as velocidades angulares  $\dot{\eta}_{sh} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}] \in \mathbb{R}^3$  e as acelerações angulares  $\ddot{\eta}_{sh} = [\ddot{\phi} \ \ddot{\theta} \ \ddot{\psi}] \in \mathbb{R}^3$ . A relação agora é entre o sistema de referência espacial e o do helicóptero, ambos com origem no centro de gravidade da aeronave.

Para realizar os controles destas variáveis, o operador dispõe de cinco entradas de comando [37], assim definidas:

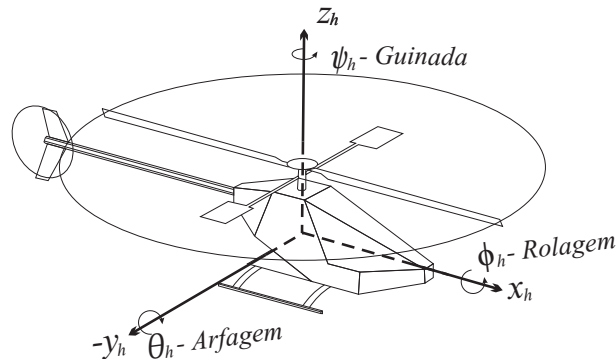


Figura 7: A representação de um helimodelo com seus graus de liberdade.

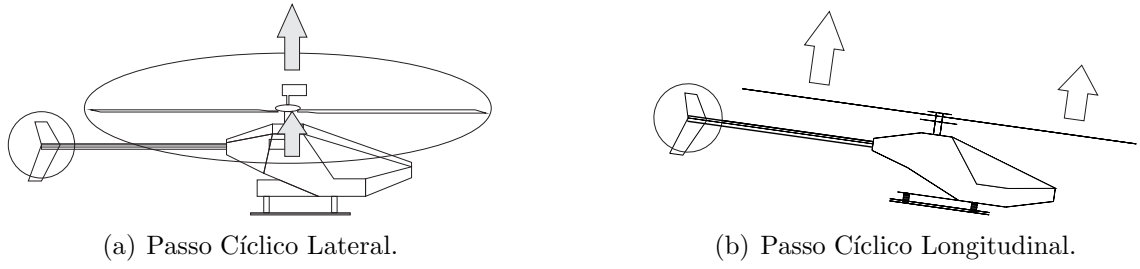


Figura 8: Passos cíclicos.

**Airelon** ( $u_{lat}$ ), que controla o passo cíclico lateral no rotor principal (ver Figura 8(a)), o qual produz o movimento de rolagem, resultando no deslocamento lateral do helicóptero;

**Profundor** ( $u_{lon}$ ), que controla o passo cíclico longitudinal no rotor principal (ver Figura 8(b)), o qual provoca a arfagem da aeronave, permitindo seu avanço (ou retrocesso) na direção longitudinal;

**Coletivo** ( $u_{col}$ ), que controla o passo coletivo do rotor principal (ver Figura 9(a)), o qual resulta no movimento de elevação, fazendo com que o helicóptero se desloque na direção vertical;

**Leme** ( $u_{ped}$ ), que controla o passo coletivo do rotor de cauda, necessário para compensar o efeito de anti-torque gerado pelo rotor principal e para produzir o movimento de guinada;

**Acelerador** ( $u_{thr}$ ), que controla a velocidade de rotação do rotor principal e do rotor de cauda. Vale comentar que no helimodelo utilizado nesta Dissertação um sistema de redução por engrenagem e uma barra de transmissão unem o rotor principal ao de cauda, mantendo fixa a relação de velocidades entre os dois. Em adição, na maioria das configurações de helicóptero, a velocidade de rotação do rotor principal é assumida constante.

Desta forma, pode-se considerar o vetor de entrada dos comandos como sendo

$$\mathbf{u} = [u_{lat} \ u_{lon} \ u_{col} \ u_{ped} \ u_{thr}]. \quad (2.3)$$

Em rádio controle a velocidade é controlada juntamente com o coletivo. Porém, para sistemas autônomos a velocidade das pás é mantida constante. Assim, tal velocidade deixa de ser uma variável de entrada, de forma que a entrada do sistema se torna

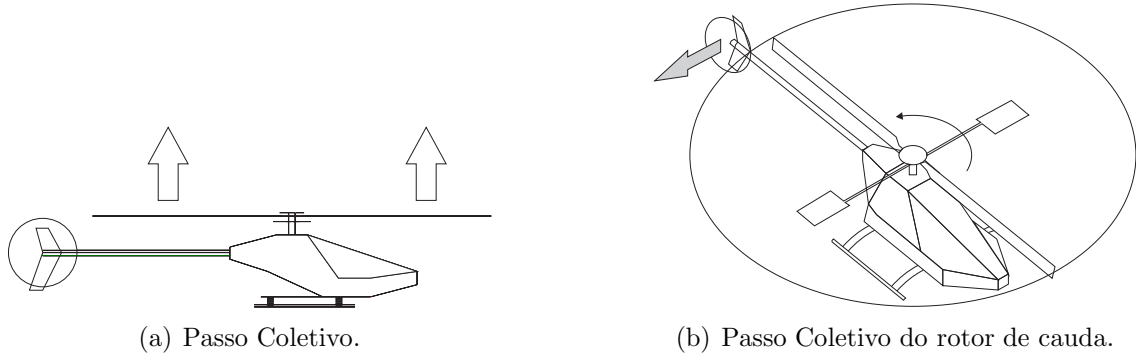


Figura 9: Passos coletivos.

$$\mathbf{u} = [u_{lat} \ u_{lon} \ u_{col} \ u_{ped}], \quad (2.4)$$

que é a entrada utilizada nesta Dissertação, enquanto as variáveis de saída são

$$\mathbf{u} = [\dot{x}_s \ \dot{y}_s \ \dot{z}_s \ \dot{\phi}_h \ \dot{\theta}_h \ \dot{\psi}_h]. \quad (2.5)$$

Pela relação entrada/saída, percebe-se que existem mais variáveis a serem controladas que comandos, o que caracteriza um sistema acoplado, que é uma característica importante de VANTs de pás rotativas. As velocidades angulares  $\dot{\phi}_h$ ,  $\dot{\theta}_h$  e  $\dot{\psi}_h$  são controladas, respectivamente, pelas entradas  $u_{lat}$ ,  $u_{lon}$  e  $u_{ped}$ , enquanto a taxa de subida e descida é controlada pela entrada  $u_{col}$ . Movimentos laterais e longitudinais  $\dot{x}_s$  e  $\dot{y}_s$  são obtidos movimentando o helicóptero de forma a produzir uma arfagem, a qual causa o deslocamento em  $x$ , ou uma rolagem, a qual causa o deslocamento em  $y$ .

## 2.2 Dinâmica dos Atuadores

Foram apresentadas nas equações anteriores as entradas e saídas de um helimodelo. Nesta seção serão apresentados em mais detalhes o funcionamento dos servo motores e do sistema de atuação, uma vez que estes são controlados pelo sistema microcontrolado desenvolvido, e a forma como eles atuam na bailarina (ou prato oscilante) do helimodelo (eles são os componentes presentes no bloco Dinâmica dos Atuadores).

Os servomotores utilizados são similares ao da Figura 10. Existem diversos fabricantes e diversos modelos, com torque, velocidade e tamanho variados. Porém, todos seguem o mesmo princípio de funcionamento e possuem características similares. O modelo da Figura 10 possui alimentação entre 4,8V e 7,4V, com torque de 6kg.cm. Para uma





Figura 10: Servomotor utilizado em helimodelismo.

rotação de  $60^\circ$  é necessário apenas  $0,040s$ , ou seja, ele gasta  $40ms$  para girar  $60^\circ$ . Vê-se que são dispositivos extremamente rápidos e precisos, pois possuem resolução de 12 bits.

Para a realização do controle, é utilizado um sinal PWM (*Pulse Width Modulation*) de período  $22ms$ , isto é  $45Hz$ . O sinal típico é mostrado na Figura 11, onde o intervalo útil de controle se situa entre  $1ms$  e  $2ms$ . Após este ponto o sinal se mantém em nível baixo.

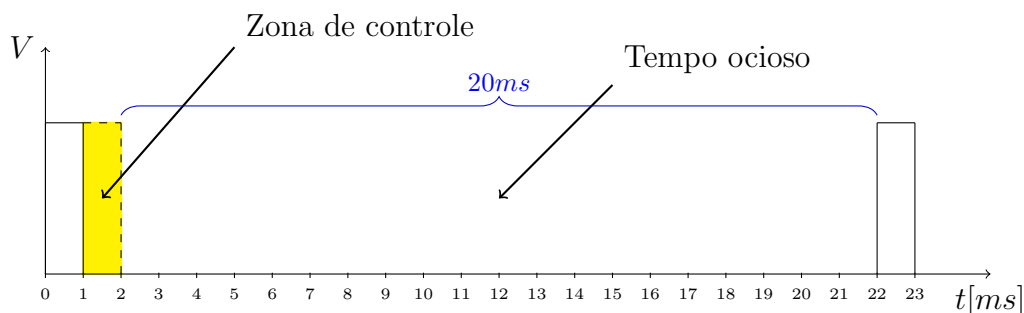


Figura 11: Sinal PWM usado no controle do servomotor.

Em helimodelos esses servos são acionados por rádio controle, onde os rádios geram um sinal PPM (*Pulse Position Modulation*) modulado em  $2,4GHz$ , que, ao ser recebido pelo receptor no helicóptero, é demodulado e decodificado em um sinal PWM para cada canal (cada servo motor). A Figura 12 mostra os dois modelos de rádio utilizados nesta Dissertação: o primeiro possui 7 canais, enquanto o segundo possui 8 canais. Cada canal transmite uma informação diferente. Este é um dos principais fatores que fazem com que um pequeno intervalo de tempo controle o movimento do servo, pois todos os canais devem estar compreendidos em uma mensagem de mesma frequência.

A Figura 13 mostra, de forma simples, como este sinal transmitido pelo rádio controle é demodulado, isto para um exemplo com 3 canais. Cada um destes canais representa uma faixa de tempo entre dois pulsos, que são transformados em sinais PWM por ordem

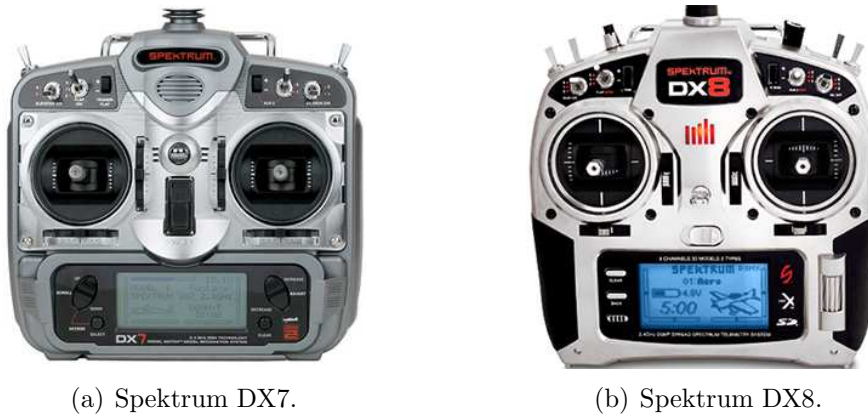


Figura 12: Rádios utilizados nesta Dissertação (fabricante: SPEKTRUM).

de chegada, como é exemplificado na Figura 13.

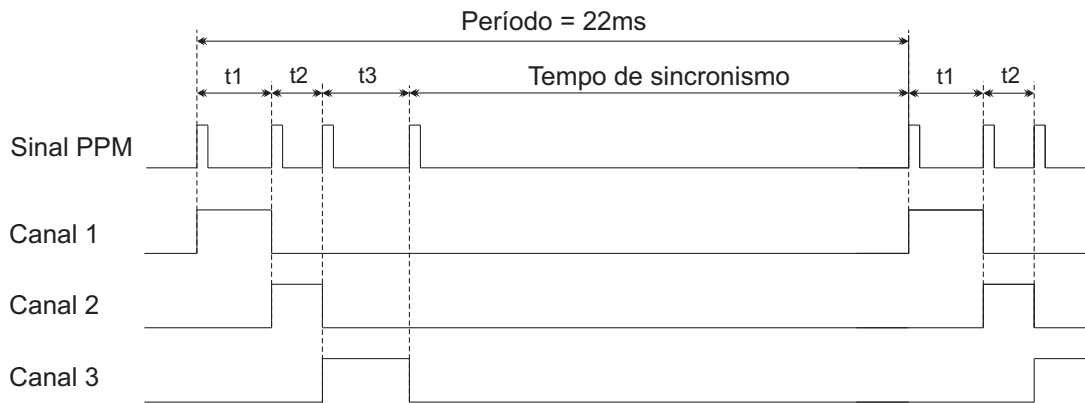


Figura 13: Separação dos canais no sinal PPM.

Antes de explicar o funcionamento do rádio controle, no que se refere aos movimentos gerados, faz-se necessário mostrar como é o sistema de atuação correspondente à parte entre os servo motores e as asas rotativas, uma vez que estas se encontram em movimento. O responsável por essa atuação é o *swashplate* ou prato oscilante (ou bailarina). Ele possui duas partes principais: uma parte rotacional, que está ligada com as pás e as barras estabilizadoras, e uma parte fixa, ligada aos atuadores ou servomotores (tais componentes podem ser visualizados na Figura 14).

Normalmente, helimodelos são dotados de barras estabilizadoras (*flybar*), cuja finalidade é amortecer o efeito de forças externas, como rajadas de vento, e suavizar os movimentos do próprio helicóptero, aumentando sua inércia, facilitando, dessa forma, a pilotagem. Estas pás respondem apenas aos movimentos cíclicos, não tendo efeito em movimentos de coletivo [48].

Como já mencionado, o *swashplate* é o responsável pela orientação da propulsão gerada pelo rotor principal. As Figuras 15 e 16 apresentam os dois movimentos mais simples,

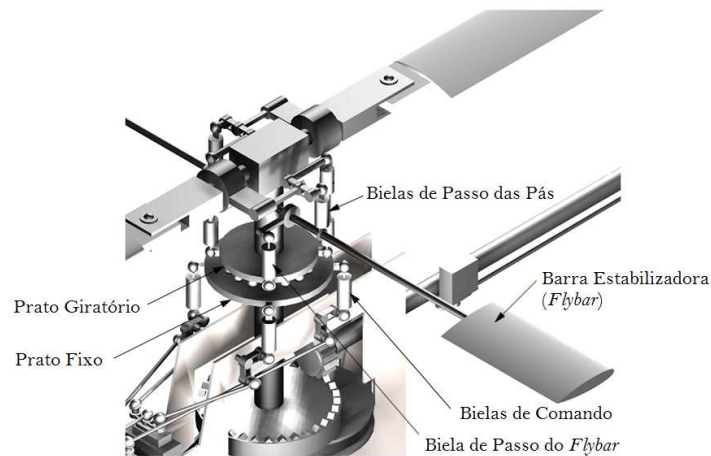


Figura 14: *Swashplate* (bailarina ou prato oscilante), responsável por converter as entradas de controle em movimento das pás do rotor principal e das barras estabilizadoras.

cujo resultado é unicamente o deslocamento do *swashplate* para cima ou para baixo em seu eixo. Note-se que o leme também possui um prato oscilante, porém este é muito mais simples, uma vez que sua única função é de alteração do coletivo.

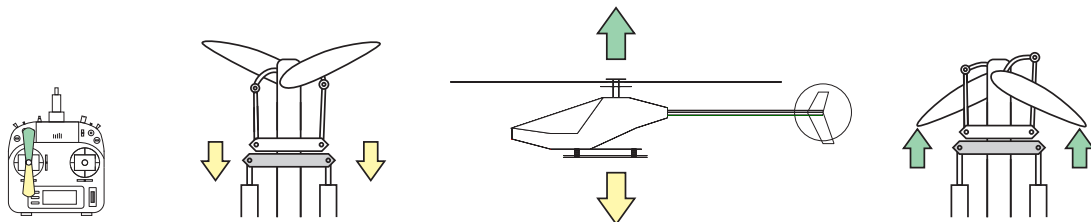


Figura 15: Diagrama do funcionamento do comando de coletivo.

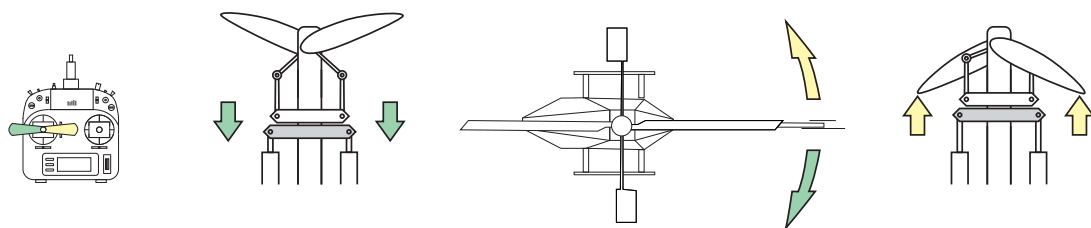


Figura 16: Diagrama do funcionamento do comando de guinada.

Em Rádio Controle o comando de aceleração é conjunto com o comando de coletivo, e um mesmo *stick* realiza os dois comandos. Pode-se ajustar o gráfico de ganho de ambos os comandos: para o caso de um sistema autônomo este gráfico é ajustado para fornecer um valor saturado de aceleração, respeitando, assim, as considerações da modelagem.

Os comandos de arfagem e rolagem também requerem o movimento conjunto do prato oscilante, porém de uma forma diferente, em relação ao coletivo. Estes podem ser vistos nas Figuras 17 e 18. Perceba-se que na Figura 17 a referência é uma vista lateral do

*swashplate* enquanto que na Figura 18 é uma vista traseira. Note-se que nas Figuras 17 e 18 são mostrados um movimento longitudinal (via arfagem) e um movimento lateral (via rolagem).

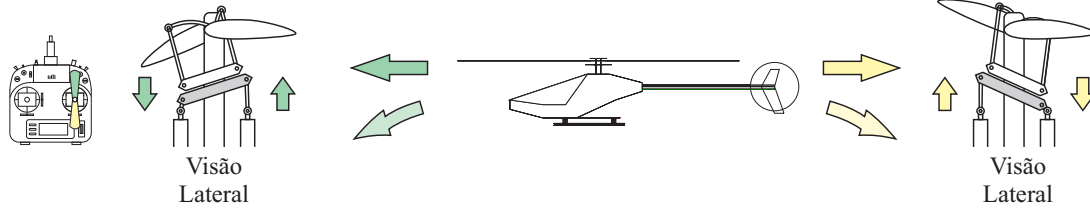


Figura 17: Diagrama do funcionamento do comando de arfagem.

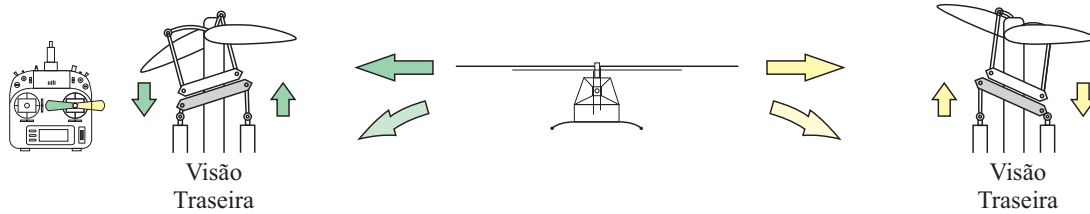


Figura 18: Diagrama do funcionamento do comando de rolagem.

## 2.3 Dinâmica das Pás Rotativas

Em relação à modelagem do veículo, estes comandos serão considerados como entradas normalizadas entre  $\pm 1$ , que serão os valores de referência para os controladores de baixo nível dos servos. Como saída do bloco Dinâmica dos Atuadores, tem-se a orientação do prato oscilante nas direções longitudinal e lateral, representadas por  $B$  e  $A$ , respectivamente. O passo coletivo do rotor principal será dado por  $\Theta_{MR}$ , enquanto o passo coletivo de cauda será dado por  $\Theta_{TR}$ . Por fim, fazendo a consideração que a saída dos servomotores é linear com respeito à largura do sinal PWM [49], as seguintes relações são obtidas

$$\Theta_{MR} = k_{1\Theta_{MR}} u_{col} + k_{2\Theta_{MR}} \quad (2.6)$$

$$\Theta_{TR} = k_{1\Theta_{TR}} u_{ped} + k_{2\Theta_{TR}} \quad (2.7)$$

$$B = k_{1B} u_{lat} + k_{2B} \quad (2.8)$$

$$A = k_{1A} u_{lon} + k_{2A}, \quad (2.9)$$

onde  $k_{1i}$  e  $k_{2j}$  são parâmetros de linearização, que dependem do modelo do helicóptero e da velocidade de rotação do rotor principal.

Considerando que a velocidade do rotor principal é constante, para obter uma variação na propulsão gerada é necessário modificar o ângulo de ataque das pás. Como já apresentado, este movimento é realizado pelo prato oscilante. Sendo assim, este ângulo é uma função que depende da posição angular das pás ao redor do eixo  $z$  do referencial do helicóptero (ângulo  $\Psi$ ), sendo descrito por

$$\Theta(\Psi) = \Theta_{MR} + A \cos \Psi + B \sin \Psi, \quad (2.10)$$

onde  $\Theta_{MR}$  é o ângulo de ataque médio determinado pelo comando de passo coletivo. Para o helicóptero em questão, assume-se que as pás giram em sentido horário e que a posição de uma pá corresponde a um ângulo  $\Psi$  igual a zero quando ela está sobre a cauda (ver Figura 20).

Segundo a Teoria de Momento, o fluxo da massa de ar através das asas tem direção oposta à força gerada. A Figura 19(a) mostra esta situação. Segundo a Teoria da dinâmica das pás, a magnitude da força de propulsão depende da configuração física das pás e de sua situação corrente. Dentre esses parâmetros, encontra-se o ângulo de ataque da pá  $\Theta(\Psi)$ , ilustrado na Figura 19. Detalhes de tais teorias fogem ao escopo desta Dissertação, mas podem ser encontrados em [50, 51, 52, 44, 53, 54, 37].

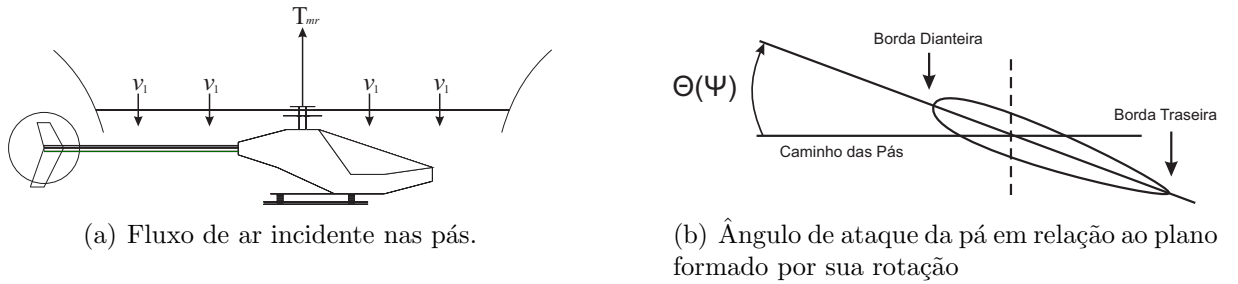


Figura 19: Ângulo de ataque  $\Theta(\Psi)$  das pás do helicóptero.

Note-se que a posição das pás nas Figuras 17 e 18 é diferente, o que se deve ao fato de que esta posição é uma função periódica do ângulo  $\Psi$ , que pode ser expressa pela série de Fourier

$$\beta(\Psi) = \beta_0 + \beta_{1c} \cos \Psi + \beta_{1s} \sin \Psi + \beta_{2c} \cos 2\Psi + \beta_{2s} \sin 2\Psi + \dots \quad (2.11)$$

Segundo [50], em helicópteros convencionais de escala real a magnitude da segunda harmônica é menor que 10% da magnitude da primeira. Desta forma, as harmônicas de

segunda ordem e de ordens maiores são ignoradas, resultando em

$$\beta(\Psi) = \beta_0 + \beta_{1c} \cos \Psi + \beta_{1s} \sin \Psi. \quad (2.12)$$

A evolução de  $\Psi$ , e a posição correspondente das pás, pode ser visualizada na Figura 20.

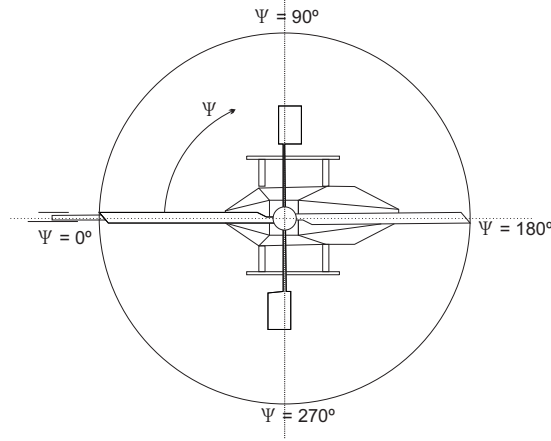


Figura 20: Posição angular das pás ( $\Psi$ ) ao redor do eixo do rotor principal.

O módulo da propulsão gerada pelo rotor principal,  $T_{MR}$ , é dado como função de  $\Theta_{MR}$ , o ângulo de ataque associado às pás do rotor principal, enquanto a orientação é dada por  $\beta_{1s}$  e  $\beta_{1c}$ . Já o módulo da propulsão do rotor de cauda,  $T_{TR}$ , é função de  $\Theta_{TR}$ , o ângulo de ataque das pás do rotor de cauda (como já comentado, não há passo cíclico no rotor de cauda, só passo coletivo, e por isto a orientação de  $T_{TR}$  é fixa, sempre na direção do eixo  $y$  do sistema de referência do helicóptero). A configuração física dos passos cíclicos pode ser visualizada nas Figuras 21(a) e 21(b).

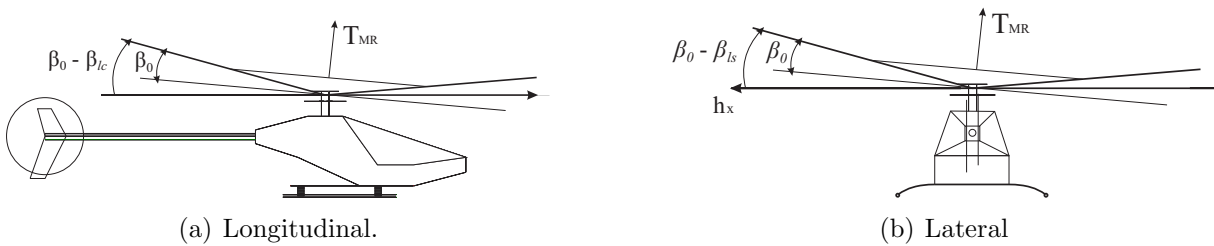


Figura 21: Ângulos que determinam a orientação da propulsão.

## 2.4 Processo de Geração de Forças e Torques

Partindo agora para o bloco de Geração de Forças e Torques, os módulos da propulsão do rotor principal e de cauda, associados às orientações  $\beta_{1s}$  e  $\beta_{1c}$ , são decompostas em

um vetor tridimensional de forças  $\mathbf{f}$  e outro de torques  $\mathbf{t}$ , que irão atuar no corpo do helicóptero.

Analisando o vetor de forças, tem-se como forças internas as propulsões geradas, e como forças externas as rajadas de ventos incidentes sobre o veículo, aqui consideradas como distúrbios, uma vez que os testes de validação foram realizados em ambiente interno e a baixas velocidades, de forma que a influência dessas forças externas é pequena, comparada à dinâmica da aeronave. Assim sendo, o vetor de forças que atua sobre o corpo do helicóptero pode ser descrito como

$$\begin{aligned} {}^b f_x &= T_{MR} \sin \beta_{1c} \\ {}^b f_y &= T_{MR} \sin \beta_{1s} \\ {}^b f_z &= T_{MR} \cos \beta_{1c} \cos \beta_{1s} \\ {}^b f_{ytr} &= T_{TR} \end{aligned} \quad \Rightarrow \quad {}^b \mathbf{f} = \begin{bmatrix} {}^b f_x \\ {}^b f_y - {}^b f_{ytr} \\ {}^b f_z \end{bmatrix}.$$

Analisando os torques tem-se o torque resultante da rotação dos rotores principal e de cauda, o torque devido ao arrasto aerodinâmico das pás do rotor e o torque devido à aceleração e desaceleração das pás dos rotores. O vetor de torques resultante, então, é dado por

$$\begin{bmatrix} {}^b L_T \\ {}^b M_T \\ {}^b N_T \end{bmatrix} = \begin{bmatrix} 0 & -H_{mz} & -H_{my} & L_{tz} \\ H_{mz} & 0 & H_{mx} & 0 \\ H_{my} & -H_{mx} & 0 & -L_{tx} \end{bmatrix} \begin{bmatrix} {}^b f_x \\ {}^b f_y \\ {}^b f_z \\ {}^b f_{ytr} \end{bmatrix}, \quad (2.13)$$

onde os termos  $H$  e  $L$  representam as distâncias cartesianas do centro de gravidade (ponto de controle) ao ponto central das pás do rotor principal e ao ponto central das pás do rotor de cauda, conforme mostrado na Figura 22, sendo tais distâncias os braços das propulsões dos rotores, que resultam nos torques aplicados à aeronave.

Para melhor entendimento do movimento de arrasto aerodinâmico, pode-se imaginar o helicóptero pairando no ar e que o piloto deseja aumentar sua altitude. Desta forma, o passo coletivo será incrementado, aumentando o ângulo de ataque das pás, gerando assim um maior arrasto aerodinâmico das pás na massa de ar, causando uma perda de momento angular das pás, que será compensada por uma aceleração do motor principal. Em contrapartida, para compensar esse aumento de momento gerado pelo sistema, o veículo tenderá a girar no sentido oposto ao de giro das pás. Por simplicidade, como mostrado em [37], o momento devido ao arrasto aerodinâmico das pás dos rotores pode ser aproximado por uma relação quadrática em função da propulsão do rotor principal e

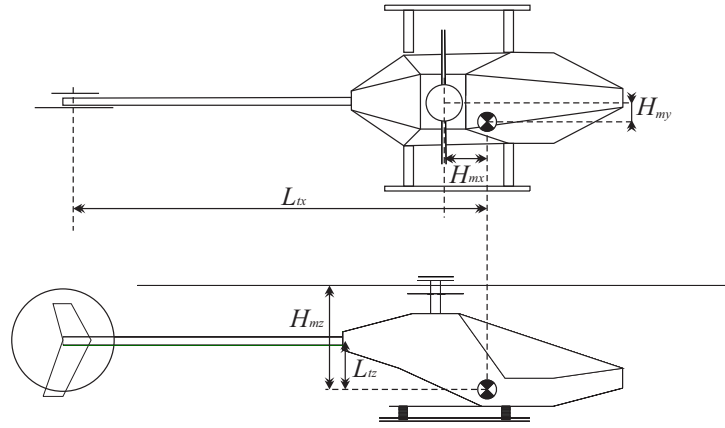


Figura 22: Definição dos braços para as forças aplicadas ao helicóptero, responsáveis pelos torques aplicados à aeronave.

de cauda, dada por

$$Q_{MR} = A_{MR}T_{MR}^2 + B_{MR}$$

$$Q_{TR} = A_{TR}T_{TR}^2 + B_{TR},$$

onde  $B_{MR}$  e  $B_{TR}$  são os momentos de arrasto das pás quando o ângulo de ataque é zero. Os coeficientes de inclinação  $A_{MR}$  e  $A_{TR}$  são uma aproximação da inclinação da curva de crescimento do ângulo de ataque das pás. Em adição, a equação quadrática que aproxima o momento aerodinâmico também inclui o momento gerado pela aceleração e desaceleração das pás dos rotores.

Assumindo que o momento gerado pela rotação das pás devido ao arrasto aerodinâmico é perpendicular ao plano formado pelo rotor principal, sua orientação será função de  $\beta_{1c}$  e  $\beta_{1s}$ , e será dada por

$$\begin{bmatrix} {}^bL_Q \\ {}^bM_Q \\ {}^bN_Q \end{bmatrix} = \begin{bmatrix} Q_{MR} \sin \beta_{1c} \\ Q_{MR} \sin \beta_{1s} - Q_{TR} \\ Q_{MR} \cos \beta_{1c} \cos \beta_{1s} \end{bmatrix}. \quad (2.14)$$

Assim sendo, o vetor de torque resultante é dado por

$${}^b\mathbf{t} = \begin{bmatrix} Q_{MR} \sin \beta_{1c} \\ Q_{MR} \sin \beta_{1s} - Q_{TR} \\ Q_{MR} \cos \beta_{1c} \cos \beta_{1s} \end{bmatrix} + \begin{bmatrix} 0 & -H_{mz} & -H_{my} & L_{tz} \\ H_{mz} & 0 & H_{mx} & 0 \\ H_{my} & -H_{mx} & 0 & -L_{tx} \end{bmatrix} \begin{bmatrix} {}^bf_x \\ {}^bf_y \\ {}^bf_z \\ {}^bf_{ytr} \end{bmatrix}. \quad (2.15)$$

Por fim, o bloco Dinâmica do Corpo Rígido é descrito pelas equações de movimento



translacional e rotacional da aeronave, quando sujeita a uma ação de forças e torques. Sua saída fornece a posição, orientação e velocidades linear e angular do helicóptero com respeito ao referencial inercial, além da taxa de variação da orientação do veículo em relação a seu próprio sistema de referência. No que se refere ao modelo correspondente a este bloco, esta Dissertação utilizará o modelo de alto nível proposto em [47], o qual não será detalhado aqui por fugir ao escopo da Dissertação.



## 3 *Instrumentação de Voo*

*“Voar, voar; subir, subir;  
Ir por onde for.”*

*(Byafra)*

Para a realização do controle automático de qualquer sistema é necessário fornecer ao controlador informações sensoriais. Em um voo livre um helicóptero possui seis graus de liberdade, os quais lhe permitem transladar e rotacionar sobre os três eixos espaciais. Durante a desenvolvimento destes movimentos a aeronave está sujeita a variações de posição, velocidade e aceleração, tanto lineares quando angulares, que somados formam um conjunto de 18 variáveis. Desta forma, o objetivo do sistema sensorial é fornecer os dados de posição e orientação do veículo aéreo em relação a um referencial global. Além da informação sensorial, é necessário desenvolver um sistema de atuação que possa realizar o acionamento dos servomotores, em substituição ao rádio controle, ou em paralelo com ele. Observe-se que originalmente a única forma de controle dos helimodelos utilizados nesta Dissertação era através de rádio controle por um operador humano.

A princípio será feita uma abordagem dos sensores que foram utilizados no projeto. Em seguida, será mostrado o estabilizador adicionado ao helimodelo para melhorar sua estabilidade em voo pairado, e por fim será apresentado o sistema embarcado microcontrolado desenvolvido especialmente para este projeto, denominado *AuRoRa Board*, que faz parte da plataforma *AuRoRa* (do inglês *Autonomous Rotorcraft Research and Development*), também desenvolvida neste projeto para simulação e realização de experimentos.

### 3.1 Unidade Inercial

A IMU (do inglês, *Inertial Measurement Unit*) é o sensor responsável por fornecer a atitude do VANT, ou seja, as variáveis de orientação (rolagem, arfagem e guinada), além de, por integração, fornecer dados de posição do veículo. Este sensor é vastamente utilizado em diversos dispositivos, estando presente em *smartphones*, por exemplo. O sensor

utilizado neste trabalho foi o  $9DOF^1$  - *Razor IMU* da *Sparkfun*, apresentado na Figura 23, que contém acelerômetro, giroscópio e magnetômetro, os quais serão apresentados abaixo, e um microcontrolador ATmega328, que é responsável pela leitura destes sensores e pela entrega dos dados de forma serial. A seguir são apresentados os sensores que compõem essa unidade inercial, e algumas de suas características mais importantes [55].

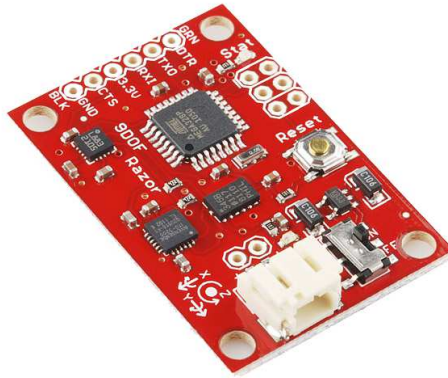


Figura 23: 9DOF Razor IMU - *Sparkfun*

**Acelerômetros:** São sensores que fornecem a aceleração linear nos respectivos eixos, sendo os mais comuns em aplicações comerciais. O modelo utilizado é o ADXL345, da Analog Devices. Ele pode ser alimentado com tensão entre  $2,0V$  e  $3,6V$ , podendo medir  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$  ou  $\pm 16g$ , com valores de 10, 11, 12 ou 13 bits, respectivamente, com consumo de corrente de  $0,1\mu A$  e taxa de amostragem superior a  $3kHz$ . Estes dispositivos são utilizados para determinar a inclinação (ângulos de arfagem e rolagem), devido à sua capacidade de medir a aceleração gravitacional. Esta funcionalidade é prejudicada em sistemas em que ocorrem altas oscilações, como helicópteros, logo apenas a medida propiciada pelo acelerômetro não é suficiente para determinar a inclinação.

**Giroscópios:** São dispositivos que medem velocidades angulares em torno dos três eixos reais. O modelo utilizado é o ITG-3200 da InvenSense, também muito utilizado comercialmente. Sua alimentação é compreendida entre  $2,1V$  e  $3,6V$ , tendo sensibilidade de 10 bits e podendo medir até  $\pm 2000^\circ/s$ , com frequência de aquisição podendo que pode chegar até a  $8kHz$ , com consumo de corrente de  $6,5mA$ . Este dispositivo apresenta filtro passa baixas programável, ou seja, é possível ajustá-lo de forma a eliminar vibrações. Desta forma, seria possível a integração numérica para determinar a posição angular referente àquele eixo. Porém, estes dispositivos apresentam o fenômeno denominado de

<sup>1</sup>Do inglês *Degrees of Freedom* (Graus de Liberdade).

*drift*, que se caracteriza por um deslize do referencial central, ou zero, ao longo do tempo, ocasionado pelo movimento do sensor ou variações de temperatura. Ou seja, supondo o sensor calibrado e estático marcando um valor nulo de velocidade angular, ao mover-se e retorná-lo ao mesmo ponto inicial o mesmo não apresentará valor nulo novamente, evidenciando a necessidade de uma fusão sensorial, uma vez que este sensor sozinho não é capaz de fornecer informações fidedignas.

**Magnetômetros:** São dispositivos capazes de medir o campo magnético terrestre, em mili-Gauss. São utilizados para a construção de bússolas digitais. O modelo utilizado é o HMC5883L da Honeywell, que possui tensão de alimentação entre 2,1V e 3,6V, 12 bits de resolução, podendo chegar a medir variações de até  $2mG$ , consumo de corrente de  $100\mu A$  e taxa de atualização de até  $160Hz$ . Sua principal função é a determinação do ângulo de guinada, uma vez que os acelerômetros são incapazes de fornecê-lo. Porém, tais dispositivos apresentam um grave problema de interferência, uma vez que são altamente sensíveis às variações magnéticas de quaisquer fontes externas, e até mesmo à proximidade de materiais ferro magnéticos. Visto a necessidade de embarcá-lo em um helicóptero, o qual possui partes metálicas e, principalmente, grande interferência devido ao motor principal, faz-se necessária uma boa calibração e a utilização de uma filtragem que leve em consideração tais pontos.

Como já citado, é necessário algum processo de filtragem para a utilização destes sensores, uma vez que cada um possui seus pontos fortes e fracos, esperando-se que, juntos, os pontos negativos sejam minimizados. Para isso é realizada uma fusão sensorial utilizando quatérnios, que esta está explicada em detalhes no Apêndice A.

## 3.2 Sensores de Referência Global

Nesta seção serão apresentados os sensores que fornecem dados de altura e posição global. Para tal, foi utilizado sensor de ultrassom e um sensor de pressão atmosférica para determinação da altura, e GPS para referência de posição global. Estes sensores são melhor detalhados a seguir.

**Barômetros:** São dispositivos capazes de medir a pressão atmosférica com alta resolução. O modelo utilizado é o BMP085 da Bosch. Sua alimentação varia entre 1,8V e 3,6V, com corrente de apenas  $5\mu A$ , e podem medir pressões compreendidas entre  $300hPa$  e  $1100hPa$ , correspondentes a altitudes entre  $+9000m$  a  $-500m$ , com resolução de  $0,1m$  de coluna de ar. Estes sensores possuem uma grande complexidade de calibração e alta

variação de valores, que podem chegar a dimensões maiores que a do próprio helimodelo, sendo necessária sua utilização em conjunto com outro sistema de medição de altura e uma filtragem adequada. A equação para determinar a altura da coluna de ar, a partir da pressão atmosférica medida, é

$$altitude = 44330 \left[ 1 - \left( \frac{p}{p_0} \right)^{\frac{1}{5,255}} \right],$$

onde  $p_0$  é a pressão ao nível do mar, de  $1013,25hPa$  [56].



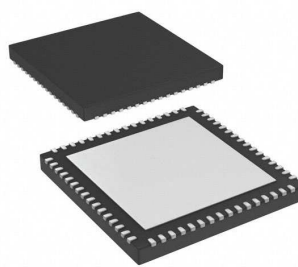
Figura 24: BMP085 - *Sparkfun*

**Ultrassom:** São sensores comuns à robótica, que têm como objetivo a medição de distâncias através da técnica de tempo de voo de uma onda mecânica de alta frequência. Foi utilizado o sensor EZ4 da MaxBotix, comprado em um circuito já montado, modelo XL-Maxsonar EZ4 da *Sparkfun*, que é mostrado na Figura 25. Ele pode operar com tensão entre 3,3V (3,4mA) e 5V (2,1mA), e possui 3 saídas distintas, uma delas via um conversor A/D de 10 bits, outra codificada em PWM e a terceira serial com 9600 de *baudrate*, tendo taxa de amostragem de até 42kHz em qualquer dos modos selecionados. No modo de saída AD, aquele aqui utilizado, o sensor fornece uma saída igual a  $(V_{cc}/1024)/cm$ , ou seja, cada bit é equivalente a 1cm, medindo um máximo de 10m e um mínimo de 20cm, segundo o fabricante. Este tipo de sensor não é recomendado para aplicações em meio externo, devido a distorções na onda em superfícies irregulares, que podem causar **difrações** na onda sonora e provocar assim erros de medida significativos. Desta forma, é recomendada sua utilização apenas em ambientes semi estruturados, como, por exemplo, superfícies planas [57].

Figura 25: EZ4 MaxBotix - *Sparkfun*

### 3.3 Microncontrolador PIC24FJ256GB206

O PIC24FJ256GB206 é um microcontrolador da Microchip, da família 24F. Possui  $256kB$  de memória de programa e  $96kB$  de memória RAM, 5 *Timers* de 16 bits cada um, 4 barramentos UART, 3 SPI e 3 I<sup>2</sup>C, além de 16 portas analógicas com conversor A/D de 10 bits, 9 canais de PWM e 9 canais de *input capture*. Possui ainda 29 portas remapeáveis, ou seja, a maioria das funções supracitadas pode ter seu porto de saída escolhido de acordo com a vontade do programador. Isto os torna microcontroladores versáteis para criação de placas de circuito impresso. Este dispositivo em específico possui 64 portas, dentre as quais 8 são para alimentação e as demais podem ser utilizadas de acordo com o necessário. Seu consumo de corrente é da ordem de  $10mA$ , com alimentação de  $3,3V$  [58].

Figura 26: Microcontrolador PIC24FJ256GB206 - *Microchip*

Dentre muitos outros, foi escolhido este microcontrolador devido a suas pequenas dimensões, diminuindo assim o tamanho final do sistema embarcado, e pelo seu baixo consumo de energia, primordial em VANTs de asas rotativas. Por fim, seu alto número de pinos *PWM*, necessários para atuação nos servos, e pinos *Input Capture*, necessários para a leitura dos comandos provindos do Rádio Controle, além de suas diversas portas de comunicação serial, o tornam versátil para introdução de novos módulos e intercomunicação

entre sistemas.

### 3.4 Computador Embarcado PicoITX

Para a realização de cálculos e o próprio controle do helimodelo foi instalado no mesmo o computador embarcado PicoITX. Trata-se de um computador com processador x86, de  $1,0GHz$ , modelo VIA C7, que possui memória RAM de  $1GB$  e periféricos iguais a um computador normal, porém com dimensões reduzidas ( $10cm$  por  $7,2cm$  e menos de  $300g$  de massa). Dentre suas interfaces vale ressaltar suas 4 portas USB e sua saída VGA. Como seu funcionamento é igual a um computador de bancada, o mesmo possui interface SATA, que neste caso é ligada a um disco rígido SSD<sup>2</sup> de  $40GB$  com Windows 7 instalado.

Juntamente a este computador está ligado um adaptador *wireless* para comunicação com um computador em terra, para supervisão ou processamento de informações extras. Para tal foi utilizado o protocolo de comunicação de rede UDP, que será explicado posteriormente.

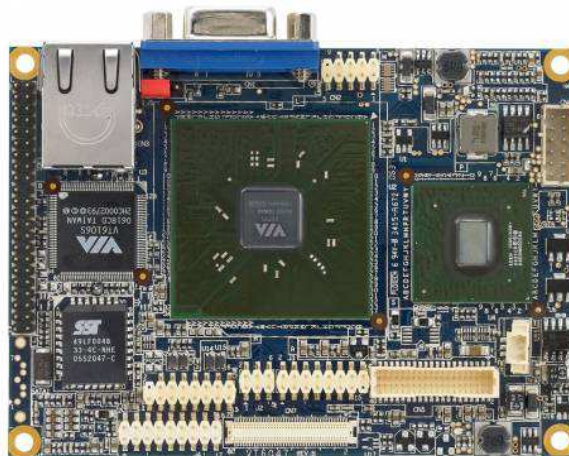


Figura 27: Pico ITX - EPIA

### 3.5 Transmissão de Dados

**UDP (*User Datagram Protocol*):** Este não é um transdutor de informação, mas sim um dos protocolos que podem ser usados quando se utiliza uma rede entre computadores, ou até mesmo pela internet. Trata-se de um protocolo simples, que é encapsulado em um pacote IPv4 ou IPv6 para ser transmitido. Porém, não há segurança

---

<sup>2</sup>Do inglês *Solid State Disk*, Disco de estado sólido.



Porta de Origem	Porta de Destino	Comprimento da mensagem	Checksum
-----------------	------------------	-------------------------	----------

Figura 28: Representação do pacote de dados do protocolo UDP.

que o mesmo chegará ao destino, ou seja, não há resposta do receptor para o transmissor confirmando o recebimento da mensagem. Isto o torna um protocolo não seguro, mas, ao mesmo tempo, um protocolo muito rápido. Em se tratando de tarefas onde o tempo é fator importante no processo, como para controladores, este protocolo é mais recomendado, devido à sua alta velocidade, em comparação com o TCP/IP, por exemplo. Para garantir a segurança do sistema toma-se certas precauções, como repetição da última mensagem caso o receptor não a receba em uma próxima iteração, e a utilização de uma rede unicamente para a comunicação entre os dois computadores em questão. Desta forma, não há carregamento da rede, evitando assim a perda de pacotes de mensagem.

A Figura 28 apresenta a formação de um pacote de dados do protocolo UDP. Para transmitir a informação para um determinado IP, deve-se informar no pacote a porta de saída do transmissor e para qual porta no receptor este pacote será enviado. O último componente *Checksum* é uma checagem para determinar se a informação é válida ou não, identificando possíveis informações corrompidas.

**Zigbee:** Para este projeto foi utilizado o módulo de transmissão sem fio Xbee da Maxstream (Figura 29). Trata-se de um transceptor que opera sobre os protocolos Zigbee e que permite a transmissão de informação entre transceptores de mesmo tipo. São dispositivos que transmitem sinais seriais TTL com alta velocidade, podendo ser superiores a  $56\text{kbps}$ , e consomem pouca energia ( $3,3\text{V}$ , com corrente de  $50\text{mA}$  durante transmissão). Estes dispositivos desempenham um importante papel na robótica, uma vez que com eles é possível obter dados do robô enquanto o mesmo realiza manobras autônomas, controlado por meio de uma comunicação sem fio, ou até mesmo realizar o controle em uma estação em terra enquanto o robô realiza apenas a atuação e captura de dados, utilizando o computador em terra no ciclo de controle (técnica conhecida como *pooling*), tudo isso com um alcance muito grande, podendo ser superior a  $1000\text{m}$  de distância em campo aberto.

Este dispositivo realiza sua transmissão através da comunicação serial, ou seja, o canal TX transmite os dados enquanto o canal RX recebe os dados, representando uma comunicação *Full-Duplex*. Ele realiza uma comunicação transparente, ou seja, após configurado não existe nenhuma necessidade de utilização de um protocolo específico. Basta enviar a

Figura 29: XBee - *Maxstream*

informação na porta RX de um X-bee que a mesma irá chegar à porta TX do outro.

**Bluetooth:** Além da comunicação Zigbee, foi utilizado como alternativa de comunicação sem fio o módulo *Bluetooth JY-MCU*, encontrado largamente para venda. Tais dispositivos são de pequenas dimensões ( $4,4\text{cm} \times 1,6\text{cm} \times 0,7\text{cm}$ ) e baixo consumo de energia ( $2,7\text{V}$ - $4,2\text{V}$  com corrente de  $8\text{mA}$ ). Eles utilizam a frequência de  $2,4\text{GHz}$  e possuem alta velocidade de transmissão, podendo ser superior a  $115200\text{kbps}$ . Note-se que são dispositivos com menos consumo que os módulos Zigbee, porém possuem pouco alcance, o qual não supera os  $10\text{m}$ . Porém, dependendo da dimensão do local do experimento ou teste que se deseja realizar, tais dispositivos são uma boa escolha para implementar a comunicação aeronave/estação em terra.

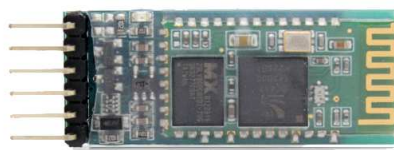


Figura 30: Módulo Bluetooth JY-MCU.

### 3.6 Estabilizador de Baixo Nível

No mercado existem alguns estabilizadores para veículos aéreos, que não têm o intuito de controlar a navegação autônoma do veículo, mas sim ajudar na estabilidade do helimodelo quando em voo pairado, e mantê-lo em uma determinada posição caso o rádio controle esteja em sua posição neutra. Neste projeto foi utilizado o Helicommand 3D<sup>®</sup>-HC, da fabricante alemã CAPTRON, que recomenda sua utilização por pilotos iniciantes, pois estes não possuem a segurança nem a perícia de pilotar, visto que pilotar helimodelos

requer bastante habilidade. Tal módulo foi instalado a bordo do helicóptero, juntamente com o controlador de alto nível e o sistema embarcado, a fim de aumentar a segurança do equipamento e dos pesquisadores, não só no caso do controle automático, como também para um controle manual, caso seja necessário, visto que nenhum dos pesquisadores é aeromodelista profissional.



Figura 31: Helicommand 3D - Captron.

O Helicommand 3D<sup>®</sup> é um dispositivo versátil, adaptável para qualquer tipo de helimodelo. Para isto deve-se configurá-lo adequadamente para o helicóptero em que ele será instalado. Essa configuração varia de acordo com o tipo do *swashplate*, devido à mixagem dos canais do rádio controle. Como já citado, o rádio controle é responsável por fazer a mistura dos seus canais para corresponder ao tipo do prato oscilante. Com o HC instalado, esta mistura de canais no controle já não é mais necessária, pois isto vira responsabilidade do estabilizador. Desta forma, o joystick é configurado com um canal por comando, enquanto o Helicommand irá fazer a tradução deste comando em movimento do conjunto de servomotores. O dispositivo que realiza isto é o Módulo RC. Na Figura 32 é apresentada a ligação dos comandos entre os dispositivos e os servomotores.

Alguns detalhes devem ser ressaltados, analisando esta figura. Um deles é o fato do canal de aceleração ir diretamente para o ESC, pois para suavizar os movimentos não é necessário o controle da velocidade por parte do Helicommand. Outro fato é que o giroscópio do helimodelo é ligado em série com o estabilizador, que também possui um giroscópio interno. Logo, deve-se ajustar o ganho de um dos dois sensores, ou remover o giroscópio do helicóptero e ligar o servo do Leme diretamente no Módulo RC, sem a ligação do giroscópio. Neste diagrama não foi dado nome aos servos, isto porque os movimentos

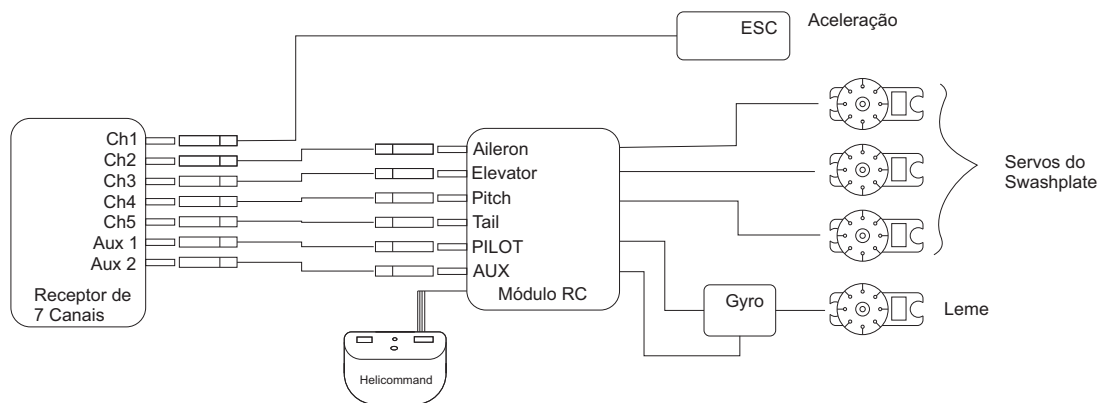


Figura 32: Conexão do módulo Helicommand ao sistema de recepção de rádio do helimodelo.

de rolagem, elevação e arfagem não são gerados por um servo independente, mas sim pelo conjunto dos 3 servos, através do *swashplate*.

O componente de controle do Helicommand é o bloco que apresenta este mesmo nome na Figura 32. É nele que os sensores estão instalados, inclusive um sensor CCD que deve ser orientado para baixo e sem obstrução do campo de visão. O detalhamento destes sensores e métodos utilizados para o processamento dos dados não é fornecido pelo fabricante.

### 3.7 Sistema Embarcado *AuRoRa Board*

Com o objetivo de realizar o controle do helicóptero, foi desenvolvida uma placa de circuito impresso específica para os helimodelos utilizados nesta Dissertação, chamada de *AuRoRa Board*. Ela faz parte da Plataforma AuRoRa, e é responsável pela captura dos dados sensoriais, como os dados oriundos da unidade inercial, do sensor de ultrassom, do barômetro e do GPS, pela leitura dos dados de entrada provindos do receptor do rádio controle, além do envio dos sinais de atuação para o Módulo RC, que os enviará para os servomotores.

A Figura 34 apresenta um esquemático da placa. O objeto central é o microcontrolador, enquanto os contatos na parte superior da placa são a entrada dos sinais de controle, à esquerda, e as saídas para o Helicommand, à direita. No canto inferior esquerdo, em cinza, estão as entradas AD para leitura das baterias. Os demais conectores na horizontal, na parte de baixo, são as três portas de comunicação serial, em amarelo, enquanto que em verde se encontra a comunicação  $I^2C$ . Entre esses, em azul, encontram-se as entradas para os sensores de ultrassom.

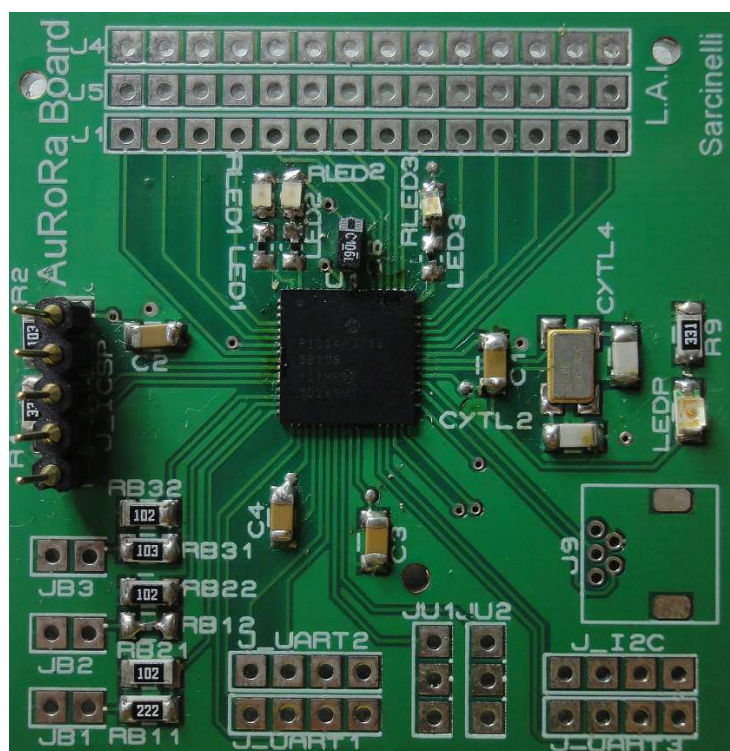


Figura 33: Imagem da placa *AuRoRa Board* (Os conectores foram omitidos para melhor visualização das placas).

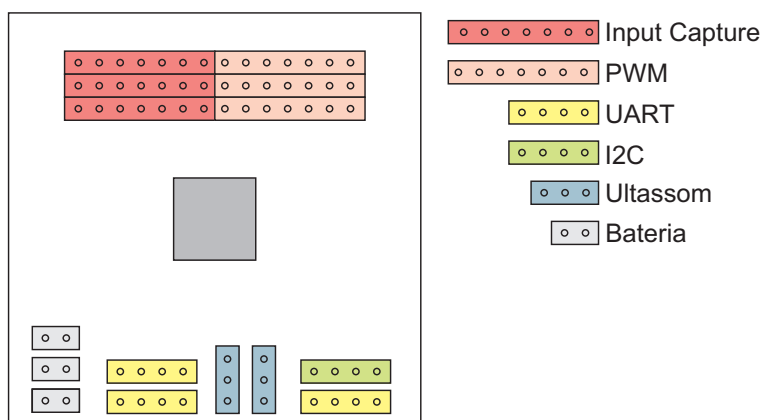


Figura 34: Esquemático da placa *AuRoRa Board*.

Deve ser frisada a versatilidade do microcontrolador utilizado: ele possui as portas remapeáveis, logo as funções supracitadas podem ter sua saída escolhida de acordo com a necessidade do programador. Isto se aplica mais para as entradas e saídas na parte superior da placa, uma vez que para o helimodelo T-REX 450 a saída se encontra à esquerda enquanto as entradas se encontram à direita. Já para o T-REX 600 é o oposto. Essa diferenciação de localização é devido à própria estrutura do helimodelo devido ao espaço restrito.

A importância da leitura das baterias deve-se ao fato da influência do nível das mesmas

no desempenho do sistema. Ou seja, para o T-REX 450 existe uma bateria, a qual é responsável pela alimentação do motor principal, dos servomotores e da placa AuRoRa Board. Devido ao alto nível de ruído gerado pela mesma, níveis baixos tendem a causar oscilações no funcionamento do microcontrolador, que não consegue realizar medidas fidedignas, o que é perigoso, pois estas medidas são a realimentação do controlador. Então, é mantido sob análise o nível de bateria. Caso fique abaixo de um nível aceitável, o experimento deve ser interrompido e a bateria trocada. Mesmo na faixa de funcionamento aceitável da bateria, o sistema apresenta respostas diferentes, ou seja, para um nível muito alto ele irá responder mais rapidamente aos comandos, enquanto para níveis mais baixos ele irá responder com um atraso um pouco maior. Então, os testes devem sempre ser realizados com a bateria com carga plena, trabalhando sempre em uma mesma faixa. Desta forma, o nível da bateria pode ser desconsiderado para análise de desempenho. Já o T-REX 600 possui 3 baterias, uma para o motor principal, uma para os servomotores e a placa *AuRoRa board* e uma para o computador embarcado. Por isso é que existem 3 entradas para leitura de bateria na placa.

Existem 3 portas de comunicação serial, pois existem dispositivos e sensores que se comunicam através dela, como, por exemplo, o módulo Zigbee, o GPS e alguns modelos de Unidades Inerciais. Não necessariamente serão utilizadas todas as portas, porém, devido à quantidade, o circuito possui uma maior versatilidade, caso seja do interesse a inclusão de algum outro dispositivo serial. A comunicação  $I^2C$  está presente pois a maioria dos chips de sensores inerciais e barométricos apresenta essa comunicação. Então, é importante a presença da mesma no circuito.

Existem diversos tipos de sensores de ultrassom, os quais podem ter diferentes formas de leitura de seus dados, como por transmissão serial, por conversão Analógico Digital e até mesmo pela geração de um sinal PWM proporcional à distância. Neste trabalho foram utilizados sensores de ultrassom com saída AD, devido à sua facilidade de leitura e pela alta quantidade destas portas nos microcontroladores. Existem duas entradas disponíveis para sensores de ultrassom, pois dependendo do modelo do helicóptero pode ser interessante a utilização de dois desses sensores para maior confiabilidade nas leituras.

Após a instalação da placa no helimodelo, o esquemático da ligação dos servomotores, juntamente com o módulo Helicommand, é o apresentado na Figura 35, onde a placa *AuRoRa Board* entra em uma posição em que é possível a leitura dos comandos do rádio controle juntamente com o envio dos comandos para o Módulo RC, que por sua vez passa a informação para os servomotores.

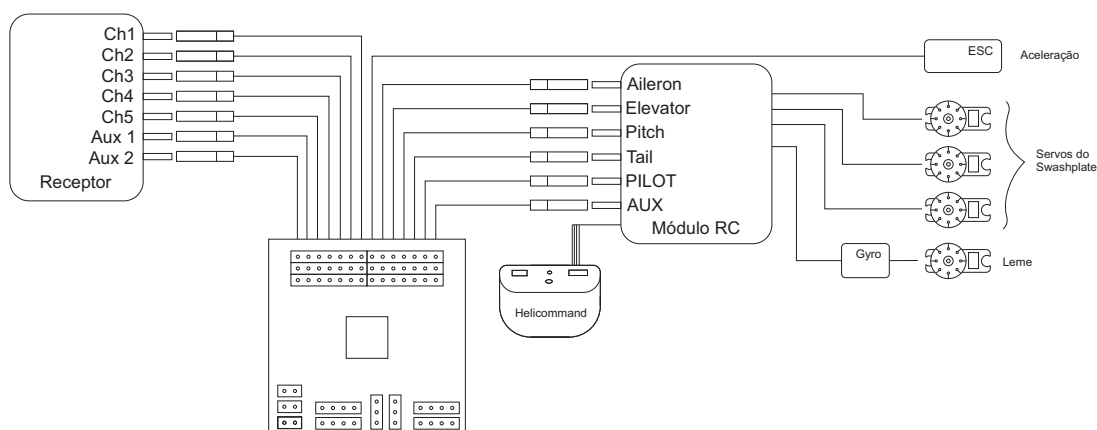


Figura 35: Ilustração da conexão da AuRoRa Board ao sistema de recepção de sinais.

Note-se que é realizada a leitura de todos os canais, inclusive da aceleração, o que se deve ao fato de que a AuRoRa board pode controlar por completo todos os canais do helimodelo, não havendo necessidade de nenhum controle do usuário pelo Rádio Controle, a não ser que isso seja de interesse do mesmo.

### 3.8 Placa de Desenvolvimento ARM

A placa de desenvolvimento da *STMicroelectronics* mostrada na Figura 36 possui um núcleo *ARM Cortex M4*, um microcontrolador com grande capacidade de processamento. Desta forma, a mesma foi adicionada ao sistema, juntamente à *AuRoRa Board*, para aumentar a capacidade de processamento embarcado. Vale ressaltar que a utilização deste núcleo ARM não exclui a placa *AuRoRa Board*, uma vez que ela realiza funções imprescindíveis para o funcionamento do sistema.

Tal placa inclui um chip contendo acelerômetro e magnetômetro, modelo LSM303 DLHC, e um giroscópio, modelo L3GD20, ambos da ST, com características similares aos sensores inerciais já citados. Como esta placa já possui estes sensores integrados, ela é uma boa opção para implementar também o filtro apresentado no Apêndice A. Desta forma, é realizada a filtragem a uma frequência de  $200Hz$ , e o envio do resultado da filtragem ao computador em  $45Hz$ . A taxa de envio foi determinada pela amostragem dos servomotores do helimodelo. Por se tratar de um filtro iterativo, sua realização em uma frequência alta garante um melhor seguimento dos movimentos e uma melhor filtragem dos ruídos. Assim, também é possível utilizar esta placa como alternativa à IMU já apresentada.

Esta placa é instalada entre a *AuRoRa Board* e o computador de controle em terra. Desta forma, todos os dados da *AuRoRa Board* são transferidos para esta placa, a qual



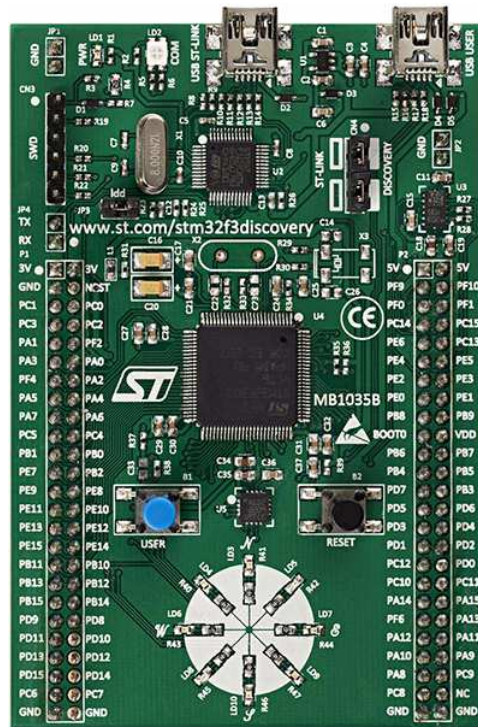


Figura 36: Placa de desenvolvimento ARM STM32F3DISCOVERY da ST Instruments.

os envia para o computador, o mesmo valendo para o fluxo das informações no sentido contrário.



## 4 A Plataforma AuRoRa

*“O segredo do sucesso é a constância do propósito.”*

*(Benjamin Disraeli)*

A utilização de simulações diminui o tempo de desenvolvimento de uma tecnologia, pois possibilita a realização de testes de forma mais rápida, às vezes menos perigosa, e menos cansativa que experimentos reais. Segundo [20], duas abordagens podem ser adotadas para desenvolver plataformas de simulação. A primeira é a abordagem *software-in-the-loop*(SIL), onde os componentes na malha de controle são todos puramente simulados, i.e, são apenas linhas de código. Este tipo de plataforma é adequado para os estágios iniciais do desenvolvimento de uma tecnologia, onde o sistema posto em teste são teorias que estão ainda sendo validadas, por meio de simulações (como exemplo, citam-se sensores, atuadores, modelo da aeronave etc. [17, 21, 59]).

A segunda abordagem é a simulação *hardware-in-the-loop*(HIL), que possui características de simulação e de experimento, sendo, um meio termo entre estes dois. Desta forma, tal simulação é entendida como um estágio mais avançado no desenvolvimento de uma certa tecnologia, um dos últimos passos antes da realização de experimentos reais, onde já existem módulos de *hardware* que podem ser colocados na malha de controle e, desta forma, serem testados em sua operação efetiva, e integrados, até que se tenha o sistema totalmente implementado em *hardware*, deixando de ser uma simulação e se tornando um experimento por completo. Ou seja, ao invés de simular os atuadores, por exemplo, estes são postos a responder aos comandos da simulação, de forma a saber se o sistema se comporta da forma prevista [60, 61, 62].

Através de simulações SIL e HIL, pode-se não somente validar um sistema e seus componentes antes de um teste experimental, como também desenvolver um *hardware* compatível com o *software* em desenvolvimento, e evitar, assim, acidentes e desgastes de equipamentos.

Neste contexto, este capítulo apresenta a plataforma de simulação/teste aqui desenvolvida, denominada *AuRoRa* (*Autonomous Robots for Research and Application*), capaz de realizar tanto simulações SIL e HIL quanto experimentos com VANTs. Ela foi desenvolvida, principalmente, como suporte para validação de modelos e controladores desenvolvidos em [47]. Com esta plataforma pode-se optar por diferentes modelos matemáticos, diferentes controladores, diferentes veículos aéreos (helimodelos ou quadrimotores), e incluir veículos terrestres e ambientes. Além de modos de funcionamento juntamente ao *hardware*. Como em [47] também há desenvolvimento de algoritmos para cooperação entre robôs, esta plataforma também dá suporte para simulações envolvendo a cooperação entre diversos veículos aéreos, até mesmo com a utilização de veículos autônomos terrestres, como em [63]. Vale mencionar que esta plataforma foi utilizada para gerar os resultados apresentados em diversos trabalhos já publicados [64, 65, 66, 23, 24, 35, 67, 36, 63, 68].

## 4.1 Estrutura da Plataforma *AuRoRa*

Antes de entrar em detalhes vale ressaltar as diferenças conceituais entre um sistema *online* e um sistema de tempo real. Um sistema é dito estar *online* com outro quando uma mudança de estado de um afeta diretamente o outro, estejam eles fisicamente conectados ou não. Por sua vez, um sistema em tempo real é aquele que garante ações e respostas em intervalos de tempo bem definidos, i.e., após ocorrer uma ação a próxima só ocorrerá após um intervalo fixo de tempo (e sempre irá ocorrer neste mesmo intervalo de tempo), sendo precedida pela resposta à primeira ação. Com base nesses conceitos, conclui-se que a plataforma aqui descrita se aproxima de um sistema em tempo real.

No Algoritmo 1 é apresentada a estrutura de funcionamento da Plataforma AuRoRa. Vê-se que é necessário uma Permissão de Execução para a realização das tarefas. Esta permissão respeita o intervalo de tempo referente ao período de amostragem de cada veículo, ou seja, caso simulado simultaneamente um quadrimotor e um helicóptero, para o primeiro veículo é respeitado o tempo de  $t_{sh} = \frac{1}{30}s$ , correspondente a seu período de amostragem, enquanto para o segundo  $t_{sq} = \frac{1}{45}s$ , cada um possuindo envio de dados de forma independente, respeitando-se, assim, tais intervalos. Desta forma, é minimizado o esforço computacional, evitando a execução das instruções de todos os veículos em um mesmo intervalo, além de evitar o envio múltiplo de sinais de controle aos veículos (em [69] é apresentado um sistema com características similares).

Observando o Algoritmo 1, é possível visualizar alguns dos módulos nos quais a

**Algoritmo 1:** Estrutura da Plataforma *AuRoRa*.

---

```

Inicialização;
while  $t < t_{max}$  do
  if Permissão de execução then
    Ler Sensores;
    Calcular posição desejada;
    if Existe joystick then
      Ler comandos;
    else
      Aplicar controlador;
    end
    Enviar sinal de controle;
    Armazenar variáveis;
  end
  if Permissão para gráfico then
    Exibir parte gráfica;
  end
end

```

---

plataforma está dividida. Estes podem estar ativos ou não, como é o exemplo da geração dos gráficos para a visualização *online* do funcionamento do sistema, ou até mesmo de um *joystick*, que pode ser utilizado por motivos de segurança (uma vez que sua atuação sobrepõe o controlador) e algum sistema sensorial (GPS, sistema de visão), tudo isto sem comprometer a execução da tarefa.

O diagrama de blocos do funcionamento interno da Plataforma *AuRoRa* pode ser visualizado na Figura 37.

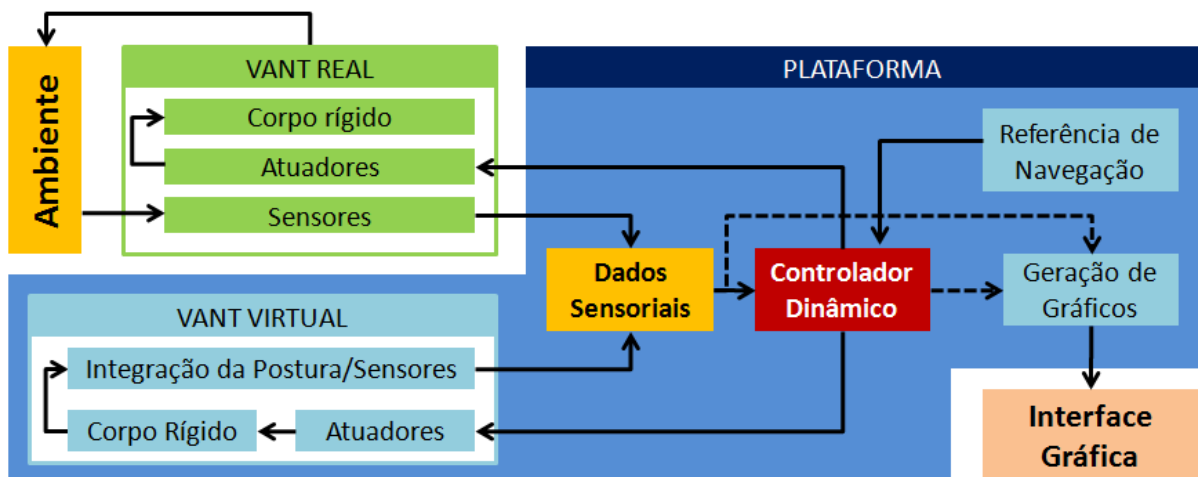


Figura 37: Diagrama de blocos do funcionamento interno da Plataforma *AuRoRa*.

Conforme mencionado anteriormente, a plataforma é capaz de simular e experimentar diversos robôs, sejam estes aéreos ou terrestres, reais, virtuais ou ambos, atuando simultaneamente, guiados pelos controladores implementados ou por *joysticks*.

Para uma descrição do funcionamento da plataforma, considere-se o quadrimotor ArDrone Parrot, o qual possui um conjunto de sensores embarcados e é capaz de gerar um *link* de comunicação *wireless* com um computador em terra, onde está rodando a plataforma *AuRoRa*. Ao se estabelecer a comunicação *wireless* (bidirecional), a plataforma inicia o processo de coleta dos dados sensoriais e envio de sinais de controle à aeronave. Entretanto, tal tarefa somente é cumprida se a *flag* de permissão de execução estiver ativada. Caso contrário, nenhuma ação é executada (i.e., os dados recebidos são ignorados e os comandos calculados são descartados). Assim sendo, em caso de ativação da permissão de execução, os dados são capturados, a referência de navegação no instante de tempo é dada e os erros de navegação são então calculados. Na sequência, com base nestes erros determinam-se os sinais de controle necessários para o cumprimento da tarefa, que, por fim, são transmitidos ao veículo.

Utilizando um VANT real, os sinais de controle são transmitidos para o sistema embarcado do veículo, que por sua vez os aplica aos atuadores. Após a reação do veículo a essas ações e sua respectiva interação com o ambiente, os sensores a bordo são excitados e registram os valores da nova condição de voo. Tais valores são transmitidos à plataforma, reiniciando o ciclo de controle. Caso se trate de um VANT simulado, os sinais de controle são enviados ao modelo dinâmico da aeronave, o qual contempla o modelo dos atuadores e o modelo de corpo rígido, além de considerar a existência de incertezas e distúrbios adicionados às manobras de voo. Após a atuação no modelo, é determinada sua postura por integração numérica, atualizando, assim, as variáveis de estado. Neste instante, tais dados ficam disponíveis para uma nova permissão de execução do controlador, que habilita a repetição do ciclo de controle. A Figura 37 ilustra esta situação.

Em ambos os casos, caso a parte gráfica estiver habilitada (uma *flag* de permissão de exibição estiver ativada), a cada período faz-se a ilustração em gráfico (ou em gráficos) do estado corrente da aeronave, ao longo da tarefa de voo. O tempo de permissão de exibição é de escolha arbitrária, e deve ter seu valor superior ao período de amostragem dos robôs envolvidos na simulação ou experimento, pois o crucial é o envio dos comandos de estabilização aos robôs. Desta forma, mostrar graficamente seu estado de navegação não deve ter prioridade, pois esta tarefa possui um custo computacional elevado, quando comparada aos ciclos de operação dos veículos.

**Nota 1** *A geração do ambiente virtual apresenta um custo computacional que pode comprometer a determinação da permissão de execução, e, assim, o envio de sinais de controle à aeronave. Portanto, caso toda a plataforma esteja rodando em uma mesma máquina, é aconselhável realizar experimentos sem permissão de exibição. Uma alternativa é distribuir as atividades de exibição e controle em dois computadores interconectados, como se detalha mais adiante, na Seção 4.2.*

Para ilustrar a questão da capacidade de processamento da plataforma, foi realizado um teste com um computador Intel®Core 2 Duo T6400, com relógio de 2.00GHz e com 4Gb de memória RAM, utilizando Windows®7 32Bits. Com este computador é possível simular, com a permissão de exibição ativada e sem perda do controle do veículo, 3 helicópteros (22ms de período), 5 quadrimotores ArDrone (35ms de período) ou 29 robôs móveis modelo Pioneer 3-DX (100ms de período). Para este último é possível simular grande quantidade de unidades porque seus controladores e sensores embarcados são mais simples que nos outros dois casos, além do seu período de amostragem ser maior. É possível, com base nesse resultado, fazer uma relação entre as demandas computacionais correspondentes a estes veículos: um helicóptero equivale a 1,6 quadrimotores, ou a aproximadamente 10 robôs móveis Pioneer 3-DX.

Por fim, vale mencionar que ao final da rotina de navegação todos os dados são armazenados, criando assim um registro, que pode ser recuperado a qualquer momento que se queira, para analisar algum detalhe do andamento da missão.

## 4.2 Estrutura Descentralizada da Plataforma

A Plataforma *AuRoRa* apresenta alguns inconvenientes no que diz respeito ao tempo de execução do laço de controle e do laço de exibição gráfica, conforme mencionado na Nota 1. Ao ilustrar a movimentação corrente dos robôs através da animação gráfica, as tarefas de envio de sinais de controle aos robôs e o recebimento de informação sensorial ficam comprometidas. Afinal, devido à estrutura sequencial do algoritmo, o sistema deve finalizar uma ação para que a próxima seja realizada.

Atentando-se a uma simulação, utilizando um helimodelo, e sabendo que a tarefa de exibição gráfica da plataforma apresenta um alto custo computacional, a Figura 40(a) apresenta os valores de tempo dos ciclos de execução (iteração), os quais variam em torno de 100ms quando é realizada a tarefa de exibição gráfica, o que superam muito o período de amostragem da aeronave. Nesta situação o VANT ficaria por um período de alguns

ciclos sem receber um novo sinal de controle, o que poderia até mesmo desestabilizá-lo.

Com o intuito de não comprometer o controle dos veículos envolvidos em um experimento, uma opção do usuário é optar pela não visualização dos dados relativos à tarefa em execução, conforme sugestão apresentada anteriormente. Entretanto, muitas vezes é necessário observar o comportamento das variáveis de postura e dos sinais de controle enviados, bem como uma representação tridimensional do estado de voo da aeronave, para avaliação do desempenho de um controlador, por exemplo. Daí, distribuir as atividades de controle e exibição entre dois (ou mais) computadores interconectados torna-se uma solução interessante.

Para resolver este problema, a Plataforma *AuRoRa*, em sua versão atual, permite a utilização de uma estação em terra, onde será realizado o experimento real ou a simulação e, via protocolo UDP, transmitir os dados para uma segunda estação terrestre, conforme ilustrado na Figura 38, a fim de distribuir o esforço computacional. Note-se que a estação terrestre A é responsável pela estabilização e navegação da aeronave, enquanto a estação terrestre B trata apenas da exibição gráfica de seu estado corrente.

Em termos de rotinas, o Algoritmo 2 apresenta esta versão mais eficiente da Plataforma *AuRoRa*, esboçada através do diagrama de blocos da Figura 39.

A fim de comparar as duas versões e, por conseguinte, justificar a implementação de um sistema multi-máquinas, a Figura 40 ilustra os tempos de iteração gastos utilizando a plataforma em suas versões centralizada e descentralizada. É possível verificar que na versão descentralizada as rotinas de controle, envio e recepção de dados são executadas em

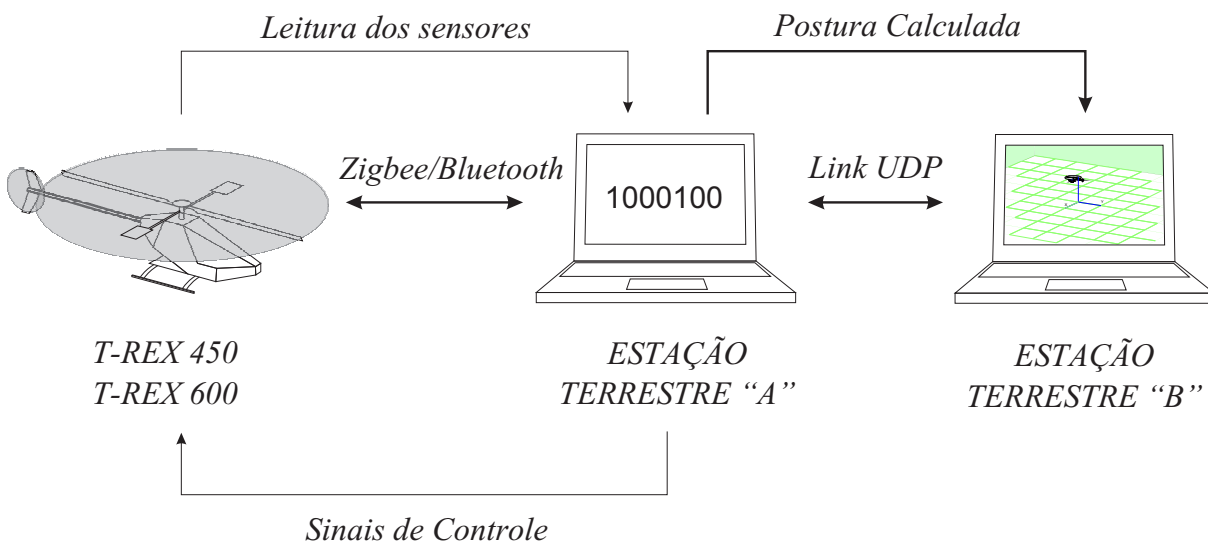


Figura 38: Estrutura da plataforma experimental com distribuição de tarefas.

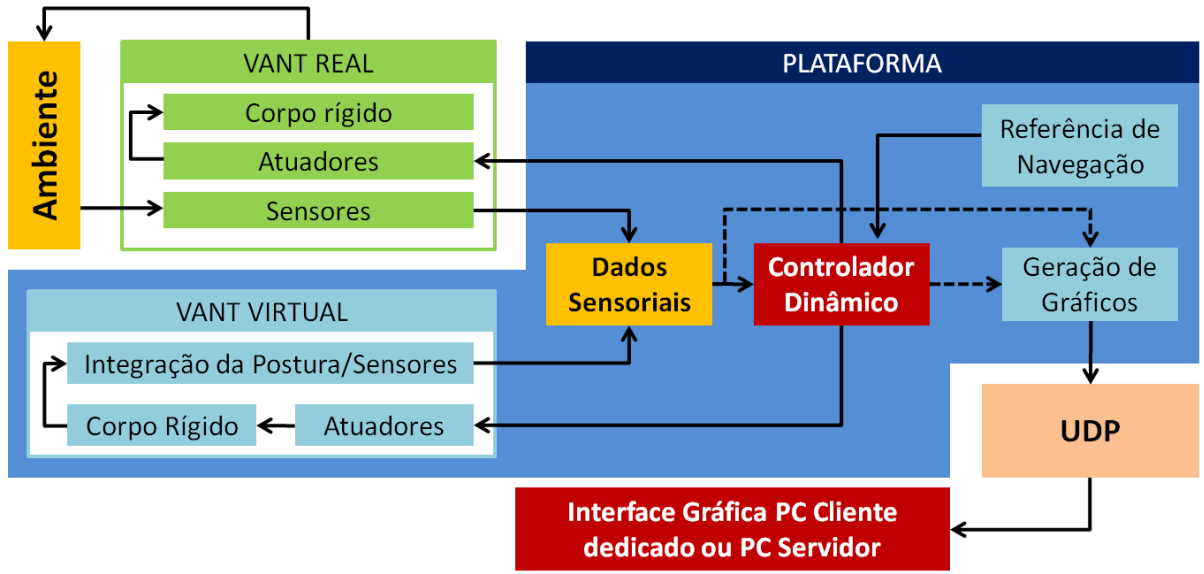


Figura 39: Diagrama de blocos do funcionamento interno da plataforma desenvolvida.

um tempo inferior a  $t_{rx} = \frac{1}{45}s$ , menor período de amostragem trabalhado, correspondente aos helimodelos T-REX 450 e T-REX 600, o que não ocorre para o caso da versão centralizada.

---

**Algoritmo 2:** Estrutura da Plataforma para múltiplos computadores.

---

```

Inicialização;
while  $t < t_{max}$  do
    if Permissão de execução then
        Ler Sensores;
        Calcular posição desejada;
        if Existe joystick then
            Ler comandos;
        else
            Aplicar controlador;
        end
        Enviar sinal de controle;
        Armazenar variáveis;
    end
    if Permissão para gráfico then
        if Permissão para Transmissão UDP then
            Exibir parte gráfica no cliente;
        else
            Exibir parte gráfica no servidor;
        end
    end
end
end

```

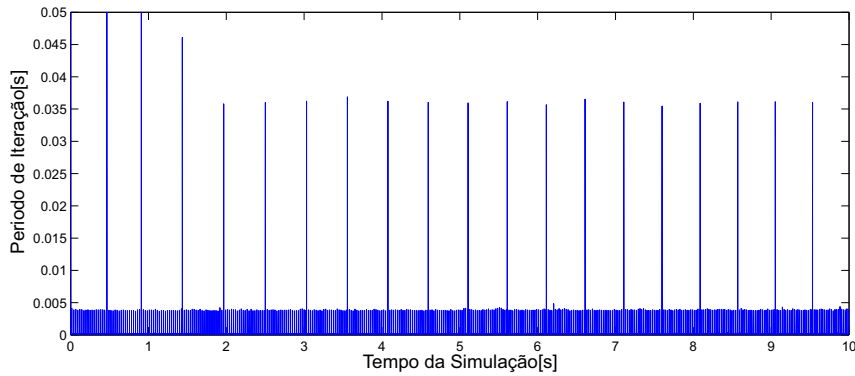
---

Na Figura 40(a) o intervalo  $\Delta i$  utilizado para a exibição gráfica dos dados é igual a  $1000ms$ , tempo suficiente para verificar uma evolução contínua da aeronave. Vale dizer que a exibição é realizada com os dados mais atuais da aeronave, dentro de tal intervalo.

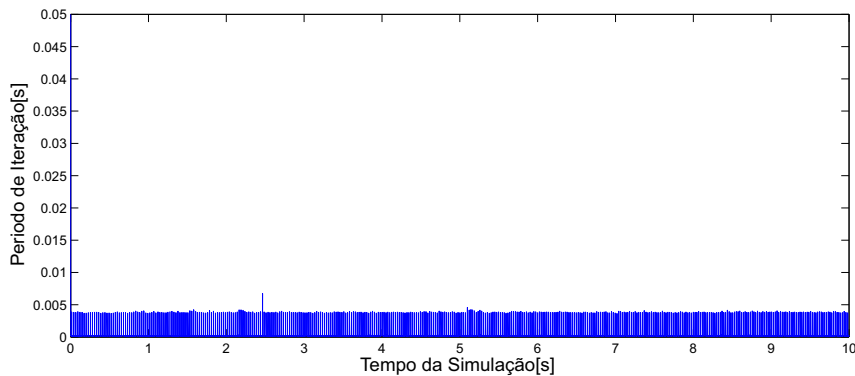
É importante mencionar que, para os testes realizados, a transmissão dos pacotes de dados para exibição gráfica em um segunda máquina utilizando comunicação UDP foi realizada em um período de tempo menor que  $1ms$ , não comprometendo, assim, o período de amostragem dos veículos. Um protocolo específico foi criado para empacotamento dos dados e desempacotamento. Em caso de perda de pacotes, o funcionamento do sistema não é afetado, pois a estação cliente simplesmente exibe o estado corrente da navegação através dos dados de entrada da comunicação UDP.

### 4.3 Estrutura Multi-veículos da Plataforma

Com relação a testes de modelos e controladores, é interessante poder simular diversos veículos ao mesmo tempo, para que, desta forma, possa ser feita uma análise comparativa entre os mesmos. Inclusive, desta forma pode-se realizar tal análise ao longo do



(a) Esquema de controle centralizado.



(b) Esquema de controle descentralizado.

Figura 40: Duração das iterações da plataforma *AuRoRa*.



experimento, através da interface gráfica que compõe a plataforma desenvolvida.

A Figura 41 apresenta a imagem do simulador em um dado instante de tempo. Nela é possível visualizar a projeção corrente da simulação em três dimensões, juntamente com variáveis em gráficos de duas dimensões, que no exemplo estão agrupados pela grandeza medida. Pode-se, ainda, optar por diferentes formas de apresentação dos dados nos gráficos de duas dimensões, sejam as variáveis correspondentes ou não ao agrupamento dos veículos. Isto pode ser selecionado ou alterado ao longo da simulação.

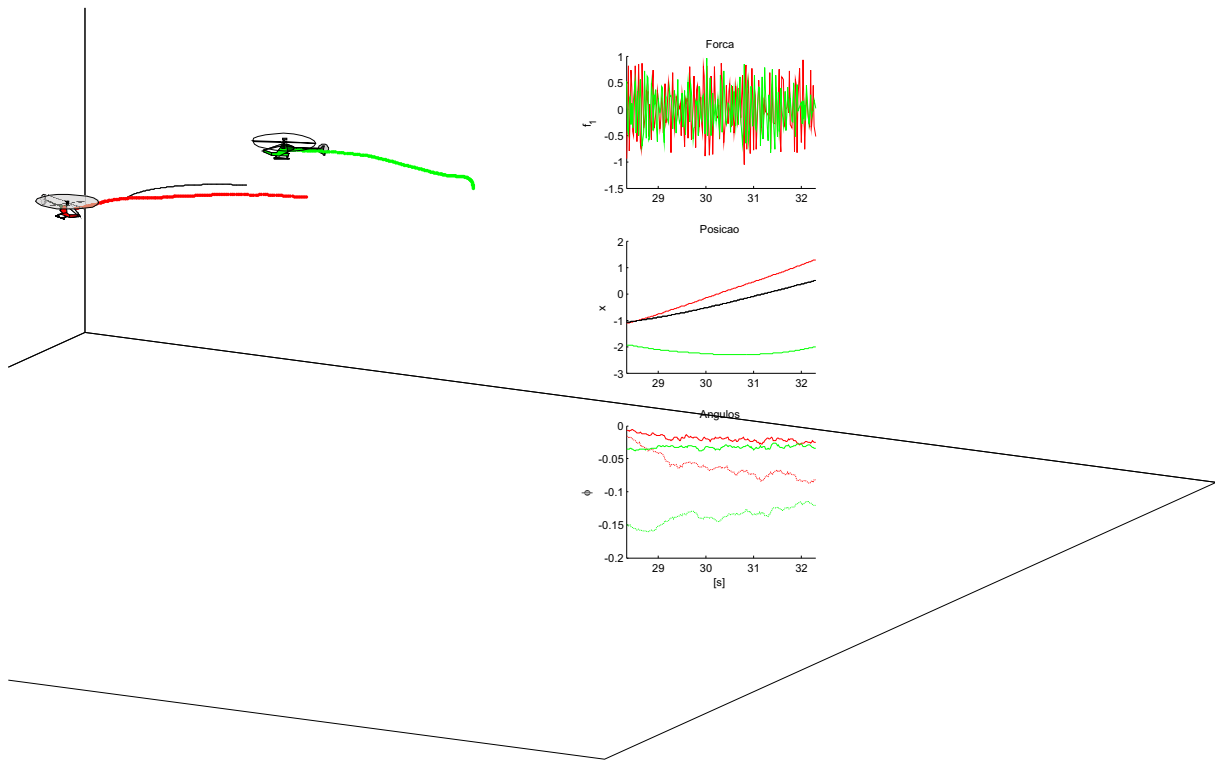


Figura 41: Apresentação de uma simulação para comparação de dois modelos.

Vale frisar o fato que a mesma interface gráfica é apresentada para o caso de experimentos reais. Desta forma, a projeção 3D do veículo é mostrada de acordo com os dados sensoriais recebidos. A simulação pode ser realizada com a quantidade de veículos que a capacidade computacional puder suportar sem comprometer o período de amostragem.

Tendo em vista a ideia da comunicação UDP e a descentralização da Plataforma *AuRoRa*, é possível não somente separar a geração dos gráficos como também separar a simulação do veículo, i.e., o computador central irá realizar apenas os cálculos relativos ao controlador e enviar os dados para os demais computadores, que serão responsáveis por gerar a dinâmica do VANT e entregar seus respectivos dados sensoriais, respeitando o período de amostragem de cada um dos veículos. A estrutura resultante é apresentada na Figura 42.

Note-se que é possível o controle de diversos veículos reais também por protocolo UDP, isto caso o helimodelo seja dotado de um computador embarcado. Caso ele não seja dotado de tal equipamento, seja por baixa capacidade de carga ou por interesse do usuário, a comunicação poder ser por meio de rede Zigbee ou Bluetooth.

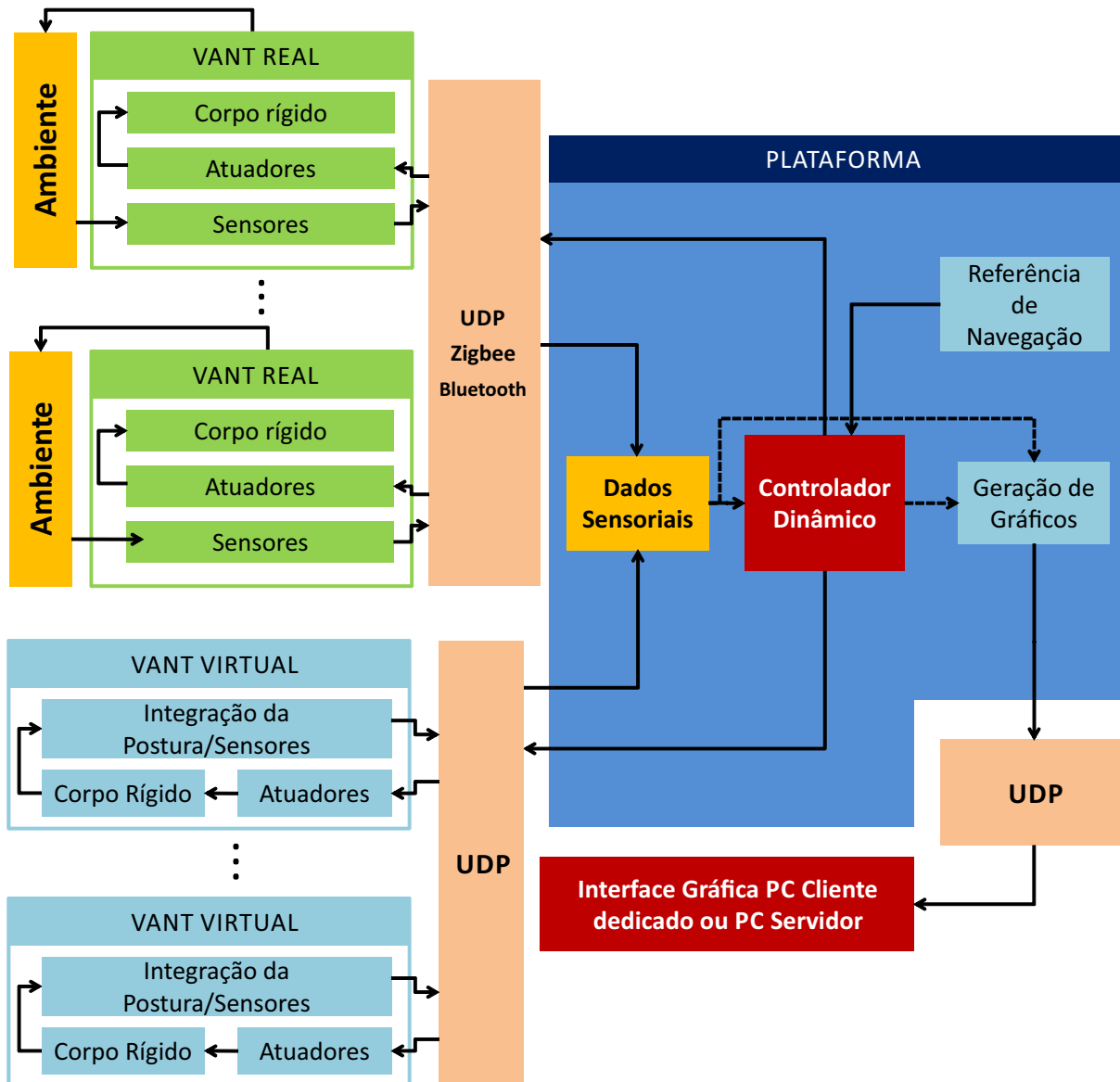


Figura 42: Diagrama de blocos do funcionamento interno da plataforma desenvolvida para multi-veículos.

Esta configuração é prevista para tarefas de cooperação entre robôs, onde um computador central com grande capacidade de processamento é embarcado em um veículo terrestre, o qual controlará os VANTs (ou mesmo embarcado em um helicóptero com maior capacidade de carga), enquanto os demais ficam mais leves para a realização de tarefas conjuntas.

## 5 *Exemplos de Utilização da Plataforma*

*“A ciência nunca resolve um problema sem criar mais dez.”*

*(George Bernard Shaw)*

Como já citado, o objetivo desta Dissertação é a criação de uma plataforma para voos autônomos de helicópteros, contemplando a criação de *hardware* e *software* específicos, para simular todo o sistema de controle e a aeronave ou para permitir ao controlador sob teste atuar sobre os servomotores da aeronave, através de sinais de controle gerados a bordo da aeronave ou em um computador em terra. Para ilustrar o uso e a versatilidade da plataforma desenvolvida, este capítulo apresenta os dados relativos a algumas simulações e experimentos reais.

### 5.1 **Simulação de Voo com Movimento de Altitude e Guinada**

Na primeira simulação apresentada, o helicóptero tem a missão de atingir uma sequência de valores distintos de guinada e diferentes cotas de altura. Desta forma, a aeronave deve realizar uma tarefa de posicionamento, contemplando manobras de ascensão e descenso, realizando um voo pairado, tendo intervalos de tempo entre as mudanças sucessivas do ponto a ser alcançado e da sua orientação.

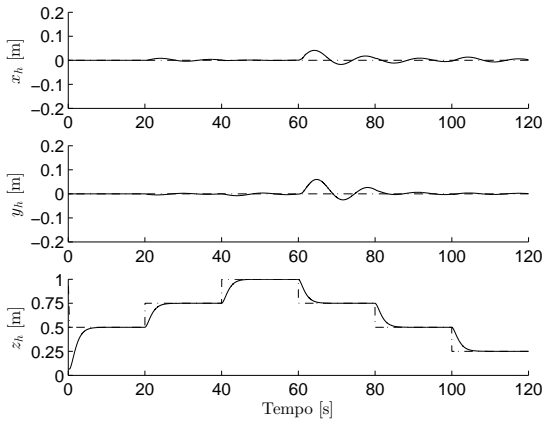
Nos gráficos apresentados, a linha tracejada representa os valores desejados, enquanto a linha contínua indica os valores efetivamente obtidos dos sensores do veículo (no caso de um experimento) ou pela integração numérica do modelo (no caso de uma simulação).

O resultado desta simulação é apresentado na Figura 43, onde são apresentados a evolução temporal do controle de altitude e guinada. De maneira mais específica, a Figura

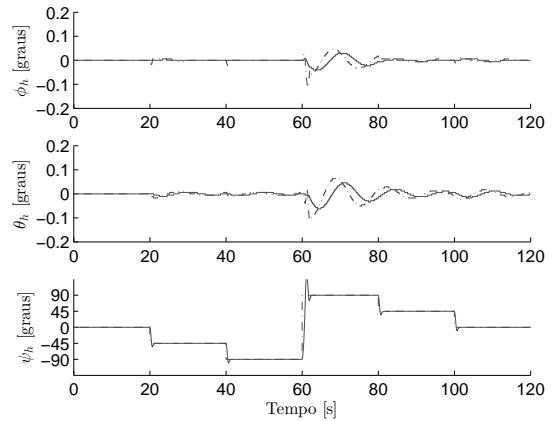
43(a) apresenta a evolução das variáveis de posição durante a simulação. Como pode ser visualizado, o controle é realizado apenas no eixo relativo à variação de altitude do helicóptero, de forma suave, atingindo todos os pontos de referência, com tendência assintótica, não havendo oscilações após atingir o valor desejado. Há pequenos deslocamentos no plano horizontal( $xy$ ), o que se deve ao alto acoplamento dinâmico entre os graus de liberdade, comuns a este tipo de aeronave, e pelas mudanças abruptas no valor desejado.

A Figura 43(b) apresenta o comportamento temporal da orientação do VANT. Observe o instante de tempo 60s, onde ocorre uma mudança abrupta de orientação, o que afeta as demais variáveis, inclusive as variações laterais e longitudinais comentadas anteriormente.

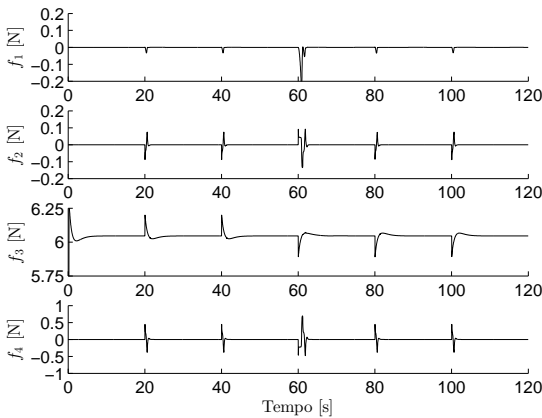
A Figura 43(c) apresenta as forças geradas para controle do helicóptero miniatura. Nota-se que a força  $f_3$  atua, principalmente, para compensar o efeito gravitacional, tendo



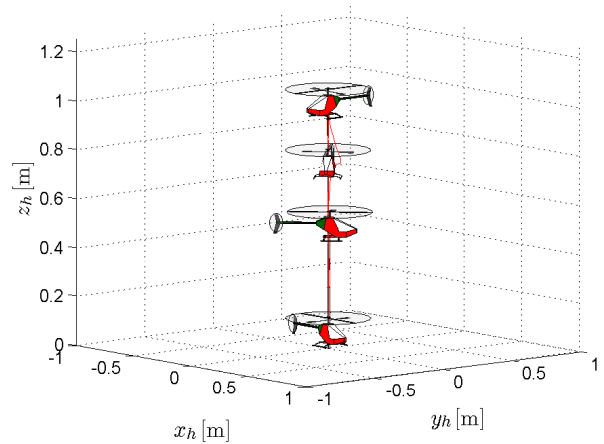
(a) Posição.



(b) Orientação.



(c) Forças.



(d) Caminho percorrido.

Figura 43: Simulação: Controle de altitude e guinada utilizando um helicóptero miniatura.

variações positivas ou negativas dependendo se o VANT deseja descer ou subir, respectivamente. Nota-se que as forças  $f_1$  e  $f_2$  apresentam pequenas variações, dado que o controle está restrito ao eixo  $z$ . Por fim,  $f_4$  apresenta variações para atingir o valor desejado de  $\psi$  e eliminar o efeito anti-torque do helicóptero, que, majoritariamente é controlado pelo giroscópio do helimodelo, portanto, para fins de controle, iguais a 0.

Para uma visualização tridimensional da simulação, é apresentada a Figura 43(d), onde os esboços do helicóptero são mostrados em intervalos de 30s.

## 5.2 Simulação de Voo Tridimensional

Nesta seção é apresentada uma simulação de um voo com helicóptero, no espaço 3D, ou seja, com controle em todos os seus graus de liberdade. Primeiramente, o veículo irá decolar verticalmente, e em seguida irá seguir a trajetória em forma de 8 em um plano inclinado, definida por  $\{x_d = 2\sin(\frac{3}{40}t), y_d = 2\sin(\frac{6}{40}t), z_d = \frac{3}{2} + \frac{1}{2}\sin(\frac{3}{40}t)\}$ , com a referência de orientação dada por  $\psi_d = 0, \forall t$ .

Nesta mesma simulação é introduzido um ruído Gaussiano de média zero e desvio padrão igual a  $0,5N$  atuando ao longo da simulação, e um distúrbio polarizado constante, de intensidade  $[0.15, -0.05, 0.02]^T N$ , atuando no intervalo  $20s < t < 40s$ .

O resultado desta simulação é apresentado na Figura 44. Note-se que antes de iniciar o movimento descrito acima, o helimodelo realiza uma manobra de decolagem, para então realizar os movimentos supracitados. Na Figura 44(a) é apresentada a evolução das variáveis de deslocamento. Os ângulos de arfagem e rolagem, necessários para a realização dos deslocamentos no plano horizontal, são visualizados na Figura 44(b), assim como a evolução temporal do ângulo de guinada. Note-se que, como citado, o veículo tem sua orientação tangente à trajetória, ou seja  $\psi_d = 0$ . A Figura 44(c) destaca as forças que são aplicadas ao helimodelo ao longo da simulação. Uma visão geral das trajetórias desejada e realizada é apresentada na Figura 44(d), que mostra a evolução da tarefa em três dimensões.

## 5.3 Leitura dos Dados de Voo

A plataforma desenvolvida tem a capacidade da leitura de todos os dados de voo do helimodelo provindos do Rádio Controle. Desta forma é possível fazer uma análise dos pontos de operação do veículo. Isso é importante para saber o comportamento do veículo

com as entradas aplicadas. Assim, é possível regular os níveis de operação do controlador para obter uma resposta mais suave ou mais abrupta, aumentando ou diminuindo a região de operação, a partir da análise dos dados de voo.

A Figura 45 apresenta os dados de um voo com o helimodelo T-REX450, realizado manualmente. Analisando a curva verde, que representa a velocidade do rotor principal, pode-se visualizar que ela cresce juntamente com o canal coletivo, como já citado em capítulos anteriores. Note-se que a velocidade cresce mais rápido que o coletivo, e estabiliza em 70%. Isto é realizado no rádio controle, onde o gráfico relativo à velocidade é limitado neste nível. Este limite é importante, porque para cada valor de velocidade existe um valor diferente de ângulo de ataque das pás, que fará com que o veículo se sustente no ar e realize seus movimentos.

Caso se deseje realizar voos com alta velocidade de manobras deve-se escolher um

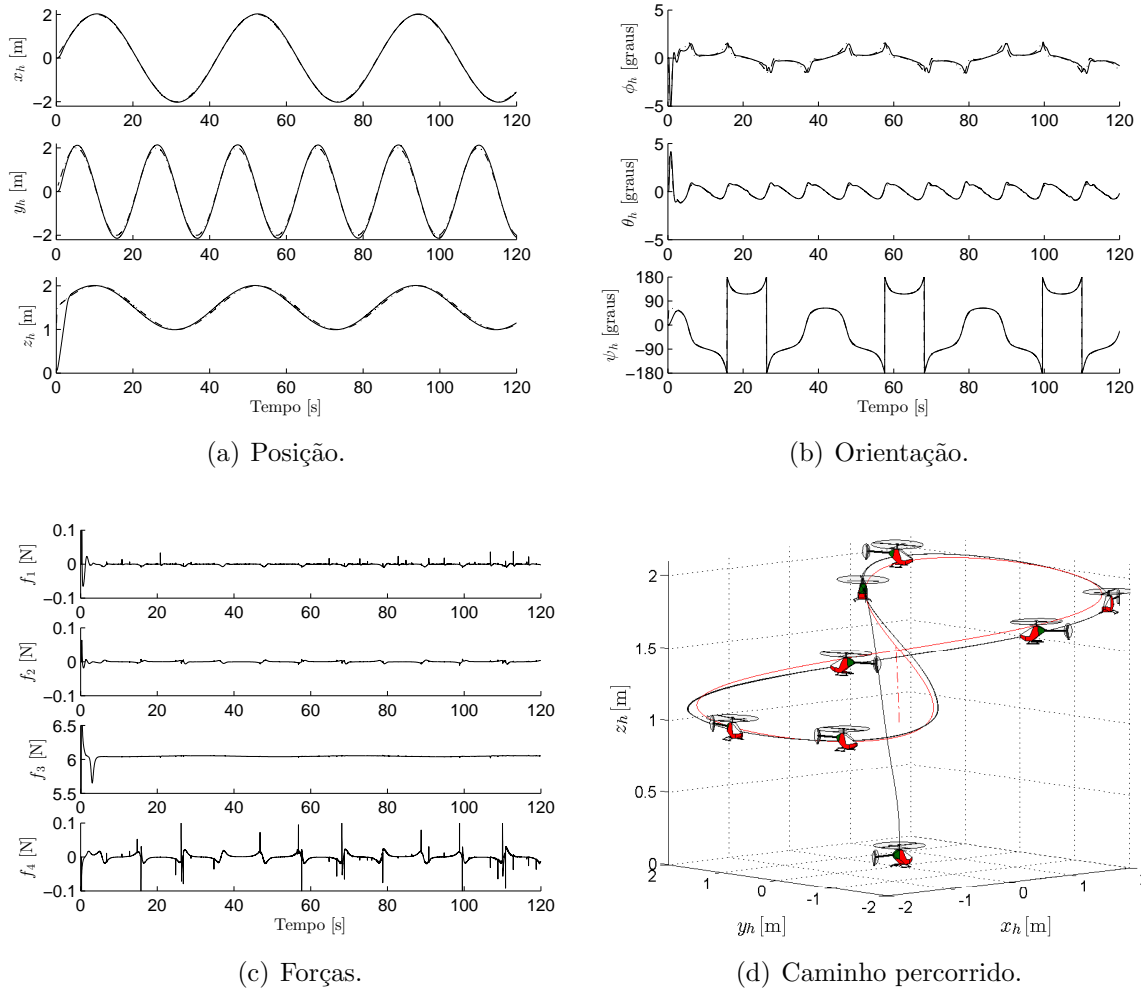


Figura 44: Simulação: Seguimento de trajetória em forma de 8 sem restrição de movimento, usando helicóptero miniatura.

valor limite alto. Caso seja necessária apenas a sustentação do VANT e a capacidade para realizar seus movimentos com estabilidade, deve-se escolher um valor baixo, porém suficiente para realizar esta sustentação, motivo pelo qual foi escolhido o valor de 70%. Vale ressaltar que as considerações realizadas no Capítulo 2 foram de velocidade constante do rotor principal; logo, independente do limite ser alto ou baixo, este deve ser constante a partir do momento em que o helicóptero decolar. O limite desta variável em um valor baixo também garante manobras menos agressivas, em comparação com aquelas realizadas a altas velocidades.

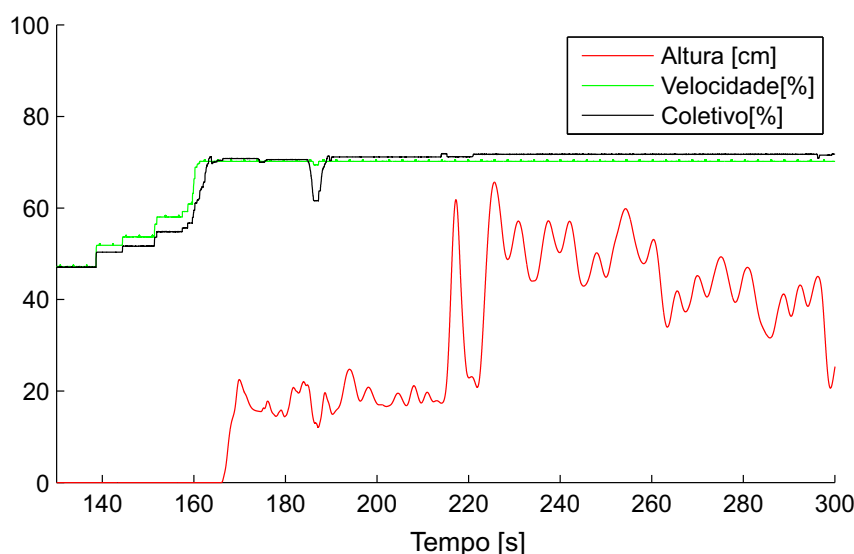


Figura 45: Leituras dos dados provindos do rádio controle do helimodelo T-REX 450.

Neste mesmo gráfico, em preto, é apresentada a evolução dos valores do coletivo. Note-se que no instante 200s houve uma pequena variação no valor, que causou uma grande ascensão. Logo em seguida este valor foi reduzido, para o helicóptero descer, pois ele estava alcançando os limites físicos do laboratório (altura do teto). Isto mostra que mesmo com a velocidade limitada o veículo ainda responde de forma agressiva aos comandos aplicados, por isso a necessidade de limitação.

A partir do instante 220s o coletivo se manteve constante, porém a altura não. A dinâmica do veículo, como citado no Capítulo 2, faz com que variações laterais mudem a orientação do vetor de propulsão, ou seja do vetor de força que garante a sustentação do veículo. Apesar de não serem mostrados os demais movimentos do VANT, o mesmo é de difícil estabilização. Desta forma, mantê-lo em uma posição fixa é de extrema complexidade, e pequenas variações no plano horizontal fazem o vetor de força mover-se, fazendo com que o veículo perca sua sustentação por algum tempo.

A Figura 46 apresenta os valores de tensão da bateria do T-REX450 durante o mesmo

voos citados acima. Vale ressaltar que, devido às dimensões do veículo, ele apresenta apenas uma bateria de 3 células, fornecendo tensões entre 11V e 14V. Pode ser visto nesta figura um alto nível de ruído, isto devido às características do motor principal, que é um motor sem escovas que pode chegar a consumir 40A. Estes ruídos podem ser prejudiciais para sistemas microcontrolados. Desta forma, a *Aurora Board* é dotada de diversos filtros, a fim de evitar que estes ruídos influenciem nas medidas. Outro fator que ajuda no isolamento é sua alimentação ser de 3,3V, bem abaixo das variações no valor de tensão da bateria.

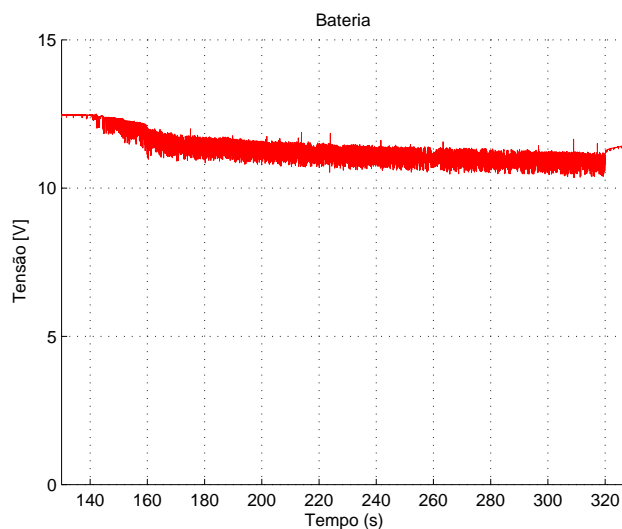


Figura 46: Valores de tensão da bateria durante um experimento.

É importante lembrar que todos os comandos do rádio controle são lidos pela placa *AuRoRa Board* e então enviados ao módulo *Helicommand*. Desta forma, tem-se o controle de todos os comandos, podendo escolher quais aplicar ou não ao helimodelo. A partir deste fato, um dos canais do controle é utilizado para escolher o modo de funcionamento do experimento, podendo este ser Manual, Semi-Automático ou Automático.

O Rádio Controle *DX7* apresenta apenas os dois primeiros modos, pois os canais de seleção deste controle possuem apenas dois estados, enquanto o *DX8* apresenta todos os três modos. No modo Manual o operador possui total controle do helimodelo, enquanto que no modo Semi-Automático o controlador automático controla a altura e/ou guinada, sendo estes selecionados por programação da *Aurora Board*. Estas variáveis foram escolhidas para compor o modo Semi-Automático pois são desacopladas das demais e sua variação abrupta não implica em desestabilização do veículo, salvo para ambientes com altura limitada, os quais podem ser perigosos. Por fim, no modo Automático o controlador atua em todos os comandos do helimodelo.

Outro fato pertinente é a equivalência entre Rádio Controle e os comandos gerados



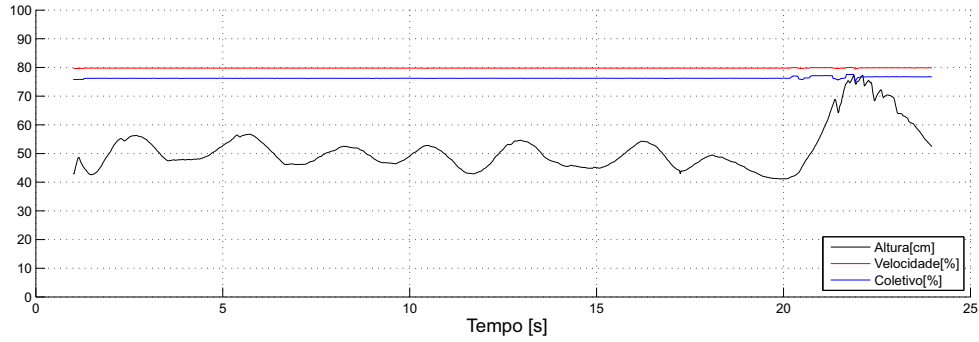
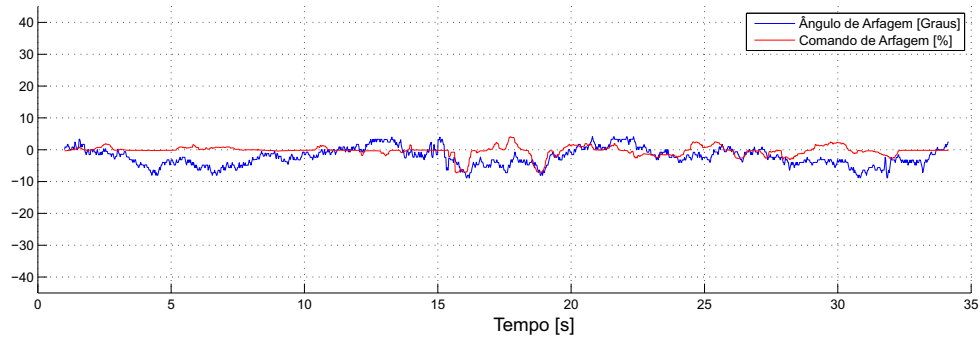
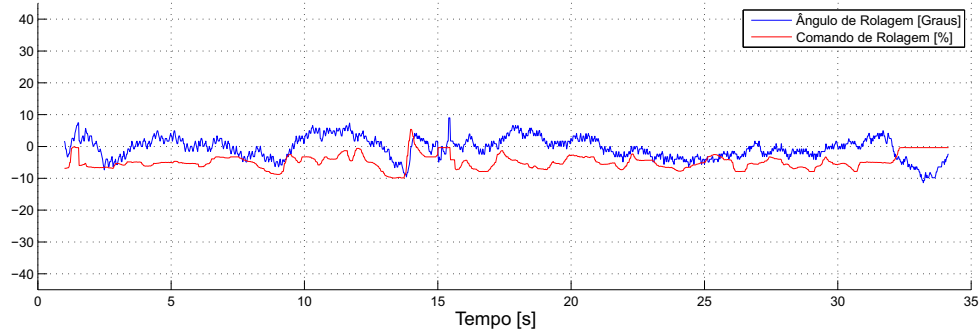


Figura 47: Valores do coletivo, velocidade e altura do helicóptero T-REX 600.

pelo controlador autônomo, pois os dois estão baseados na mesma escala, sendo que a *AuRoRa Board* foi desenvolvida para ter comandos idênticos ao do Rádio Controle. Desta forma pode ser realizada a comutação entre os dois sem causar mudanças abruptas que possam causar desestabilização do veículo por parte do piloto ou do controle automático.



(a) Ângulo de arfagem com o comando de profundor.



(b) Ângulo de rolagem com comando de aileron.

Figura 48: Experimento com T-REX 600.

Do mesmo modo que a telemetria foi testada em associação com o helicóptero T-REX 450, ela também foi testada em associação com o helicóptero T-REX 600. Um teste similar àquele sintetizado na Figura 45 foi realizado com o helicóptero T-REX 600, agora medindo os ângulos de arfagem e rolagem, usando a IMU, e seus resultados são mostrados na Figura 47. Note-se que o comportamento dos helicópteros é similar, porém, ocorre

uma maior amplitude nos movimentos de verticais para o helicóptero menor, enquanto o maior se mantém mais estável no ar.

Em outro experimento realizado, foi feita a captura dos comandos do rádio controle e os ângulos medidos pela unidade inercial. Estes podem ser visualizados nas Figuras 48(a) e 48(b). Nelas é possível reparar que existe um atraso entre o comando fornecido e o ângulo resultante. É possível notar também que mesmo pequenos valores aplicados realizam consideráveis variações angulares, evidenciando a alta sensibilidade do sistema.

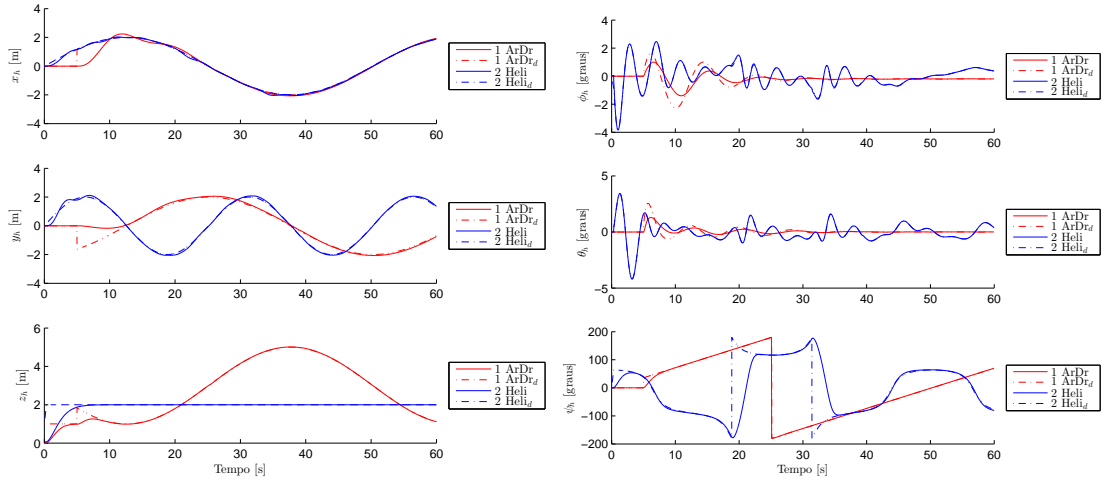
## 5.4 Simulação com Vários Veículos

A fim de testar a funcionalidade multi-veículos da plataforma, foi realizado um experimento simulado utilizando dois VANTs, um quadrimotor e um helicóptero, representados pelas linhas vermelhas e azuis, respectivamente, nos gráficos da Figura 49. O primeiro dos dois VANTs realiza um seguimento de trajetória circular em um plano inclinado, enquanto o segundo realiza o seguimento de uma trajetória correspondente a um perfil em forma de 8, no plano horizontal, com cota vertical fixa. Considerando que as linhas tracejadas representam as trajetórias desejadas, pode-se verificar que ambos os veículos executam a missão de voo de forma satisfatória, sem apresentar oscilações ou atrasos. O comportamento dos veículos pode ser observado através do caminho navegado (Figura 49(c)), bem como através da análise da evolução das variáveis de postura (Figuras 49(a) e 49(b)).

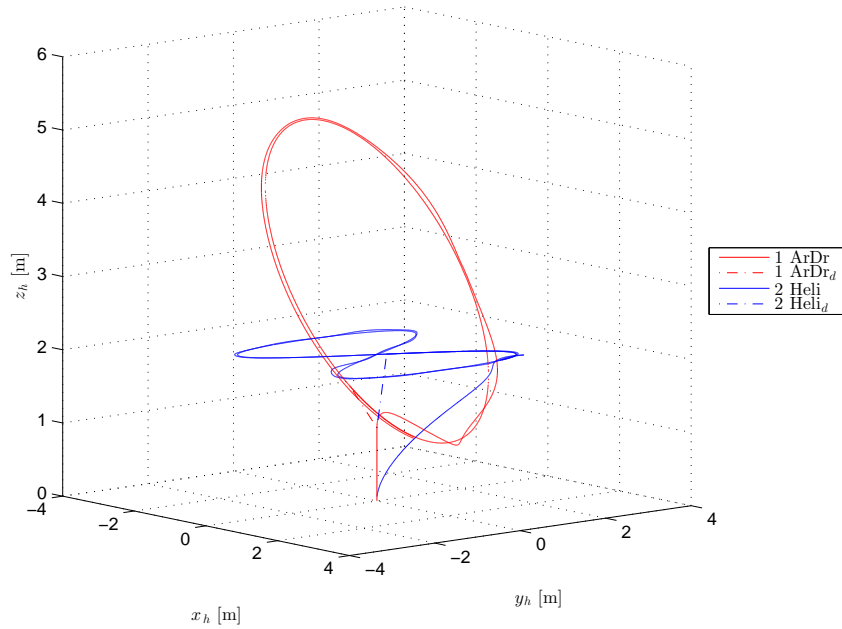
## 5.5 Simulação *Hardware-in-the-Loop*

A plataforma possui a capacidade de testar componentes separados do helimodelo. Por exemplo, após implementar a filtragem dos sinais fornecidos pelos sensores inerciais, a unidade inercial pode ser testada utilizando a plataforma *AuRoRa*. Desta forma pode-se analisar os resultados e, se eles forem satisfatórios, realizar testes com a unidade inercial instalada no VANT.

Na Figura 50 é mostrado um experimento deste tipo. As variações na unidade inercial devem, primeiramente, alterar o ângulo do veículo e ter a capacidade de acompanhar o movimento mesmo em variações mais abruptas. Na Figura 50(a) pode ser vista a variação angular dos dados. Vale lembrar que estes movimentos foram feitos manualmente, sem seguir um perfil pré-estabelecido, pois o objetivo é analisar a resposta do controlador a



(a) Evolução temporal das variáveis de posição. (b) Evolução temporal da atitude da aeronave.



(c) Trajetória dos veículos no plano 3D.

Figura 49: Simulação de uma tarefa que envolve a coordenação entre dois VANTs.

estas variações angulares.

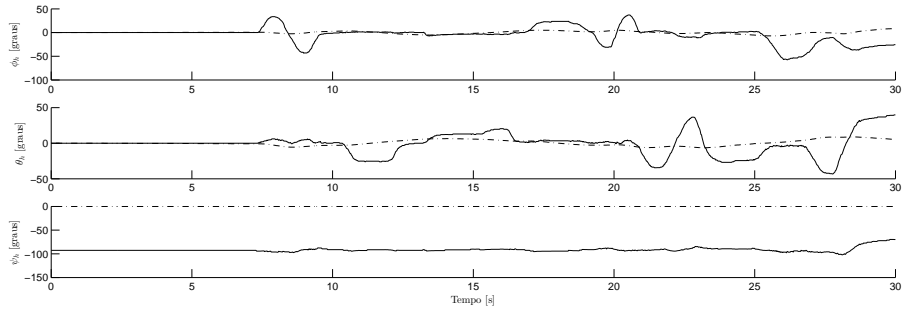
Já Na Figura 50(b) é mostrada a variação das coordenadas espaciais resultante da integração numérica dos dados provindos da unidade inercial. Analisando esta figura, vê-se que a variação está de acordo com as variações angulares do veículo. Lembrando que a variável  $z$  foi mantida constante pois as variações dos valores da IMU não implicam em variações do eixo vertical.

Por fim, analisando a Figura 50(c), onde estão mostradas as forças aplicadas pelo controlador no helimodelo, percebe-se que o controlador aplicou forças em direções opostas ao movimento a fim de tentar levar o VANT ao valor desejado, que neste caso era de 0

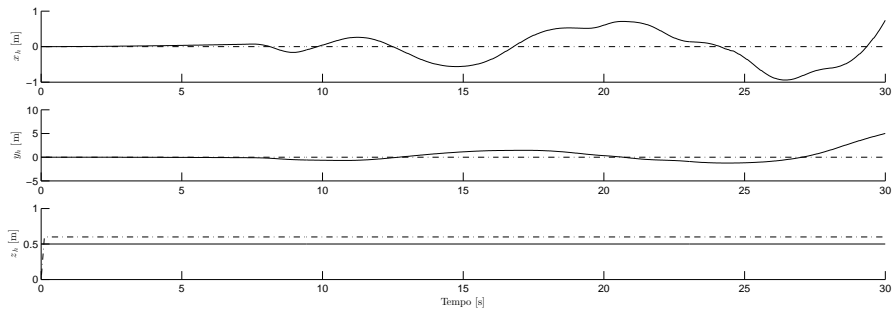
para todas as variáveis.

A analisar pela força  $f_4$ , nota-se que ela deixou de ser 0 como nas demais simulações, isso pelo fato que o ângulo  $\psi$  se manteve diferente de 0, aproximadamente em  $-100^\circ$ . Desta forma o controlador teve de aplicar uma força para tentar zerar este ângulo. O valor desta força ainda seguiu a variação do ângulo  $\psi$ . Outro fato que pode ser analisado é a saturação do controlador, que ocorreu quando a distância entre o valor desejado e o valor real era grande.

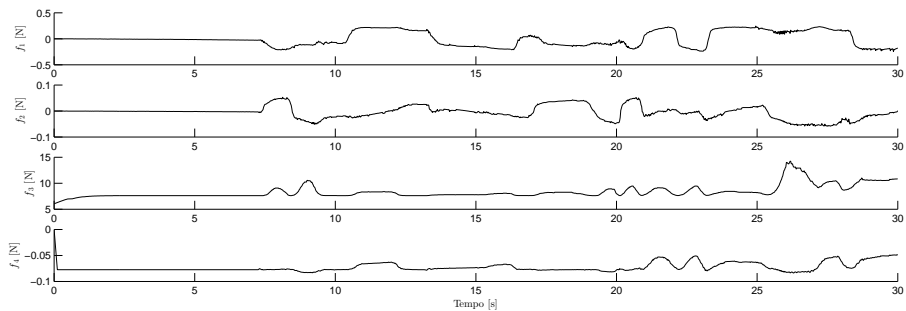
Foi utilizada uma comunicação *Bluetooth* para a comunicação com os sensores inerciais, o qual realizava uma filtragem dos dados a  $200Hz$  e um envio dos dados a  $45Hz$ . Como este é um filtro iterativo, esta configuração aumenta a precisão, não sendo necessário



(a) Ângulos.

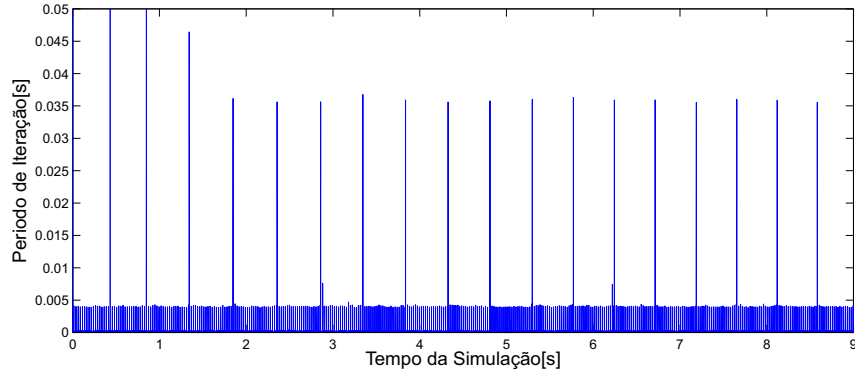


(b) Variáveis de posição.

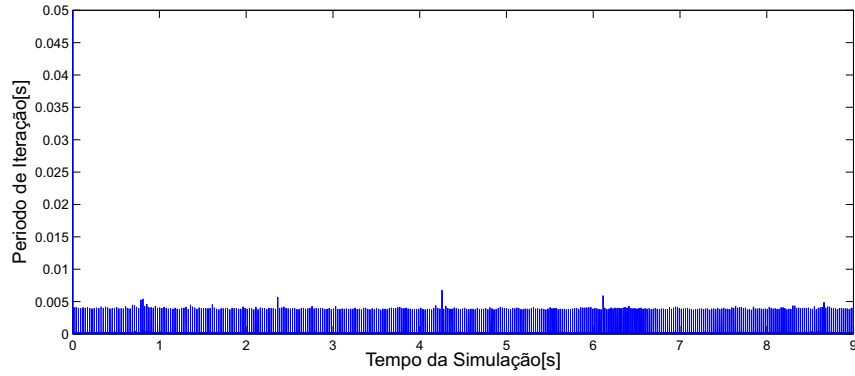


(c) Forças aplicadas pelo controlador.

Figura 50: Simulação HIL para teste da unidade inercial.



(a) Com a permissão de execução dos gráficos habilitada.



(b) Com a permissão de execução dos gráficos desabilitada.

Figura 51: Tempo de execução das iterações com a comunicação *Bluetooth*.

capturar os dados com taxa maior que  $45Hz$ , pois esta é a frequência de atualização do controlador.

Além de poder testar a filtragem, neste mesmo teste foi possível testar a comunicação. Da mesma forma que foram apresentados os gráficos relativos ao tempo de execução das iterações da plataforma no Capítulo 4, o mesmo foi realizado utilizando um experimento real com a comunicação *Bluetooth*. A *string* de dados enviada se constituía dos dados do sensor de ultrassom, do nível de tensão da bateria, dos valores dos acelerômetros e giroscópios, juntamente com os quatérnios resultantes da filtragem. O resultado pode ser visualizado na Figura 51. Ali, a Figura 51(a) apresenta os dados com a permissão de exibição dos gráficos habilitada, enquanto 51(b) apresenta os dados com a permissão de exibição dos gráficos desabilitada. Note-se que o resultado foi equivalente ao resultados da simulação, mostrando que a comunicação é capaz de fornecer os dados a uma boa velocidade sem comprometer o tempo de execução de cada iteração de controle.

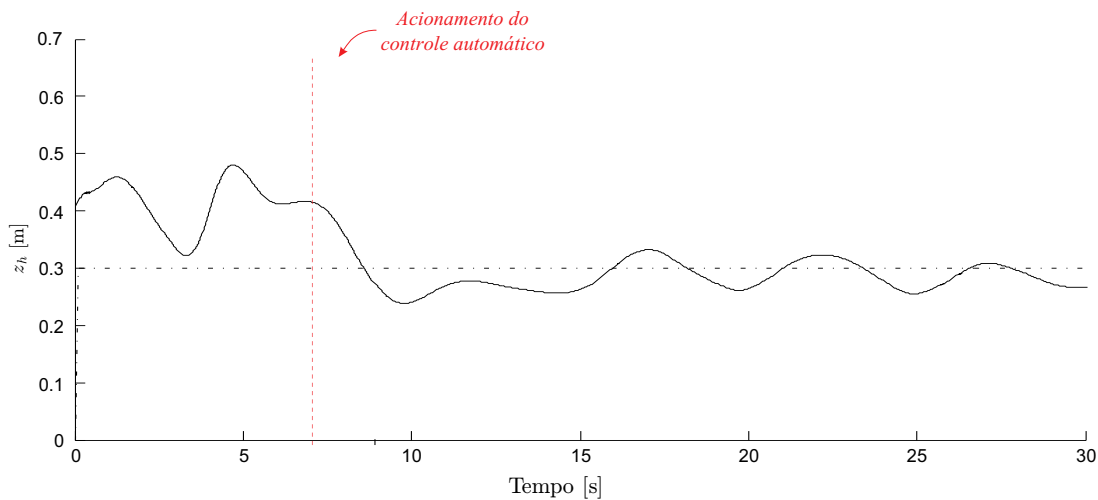


Figura 52: Controle autônomo da altura.

## 5.6 Controle Automático de Altura

A Figura 52 apresenta o resultado para um voo autônomo utilizando apenas o controle de altura. Nesta tarefa o veículo tem como objetivo manter uma cota vertical fixa, por um dado intervalo de tempo. A curva mostra que, apesar de oscilar sobre o valor, o helimodelo conseguiu se aproximar do perfil estabelecido. Note-se que para este experimento foi utilizada uma altura baixa, o que é prejudicial, pois para baixas altitudes o veículo apresenta o efeito solo, que é um efeito aerodinâmico onde o fluxo de ar gerado pelas pás do rotor principal é interrompido pelo solo, aumentando a força de propulsão no veículo nesta região, tornando-o mais suscetível a movimentos abruptos [70].

Como citado anteriormente, por questões de segurança, o rádio controle é o responsável por habilitar ou não os comandos do controlador autônomo. Desta forma, após a inicialização do experimento o operador deve escolher um momento para habilitar o acionamento dos dados vindos do controlador. Se for necessário, pode-se desabilitar o controle autônomo, e o helimodelo volta ao controle manual. Os instantes de tempo nos quais é habilitado ou desabilitado este modo de funcionamento são indicados nos gráficos a seguir.

Uma missão mais elaborada foi realizada nos experimentos seguintes, onde o veículo deve estabilizar em 30cm, depois alterar sua altura para 60cm e, após 30 segundos, retornar e estabilizar novamente em 30cm. Na Figura 53, após o acionamento do controlador, claramente vê-se um sistema instável, o qual precisa ter seus ganhos ajustados.

É realizado, então, um ajuste dos ganhos, porém desta vez utilizou-se um valor maior para altura, para ficar fora da região sobre influência do efeito solo. Assim, os valores

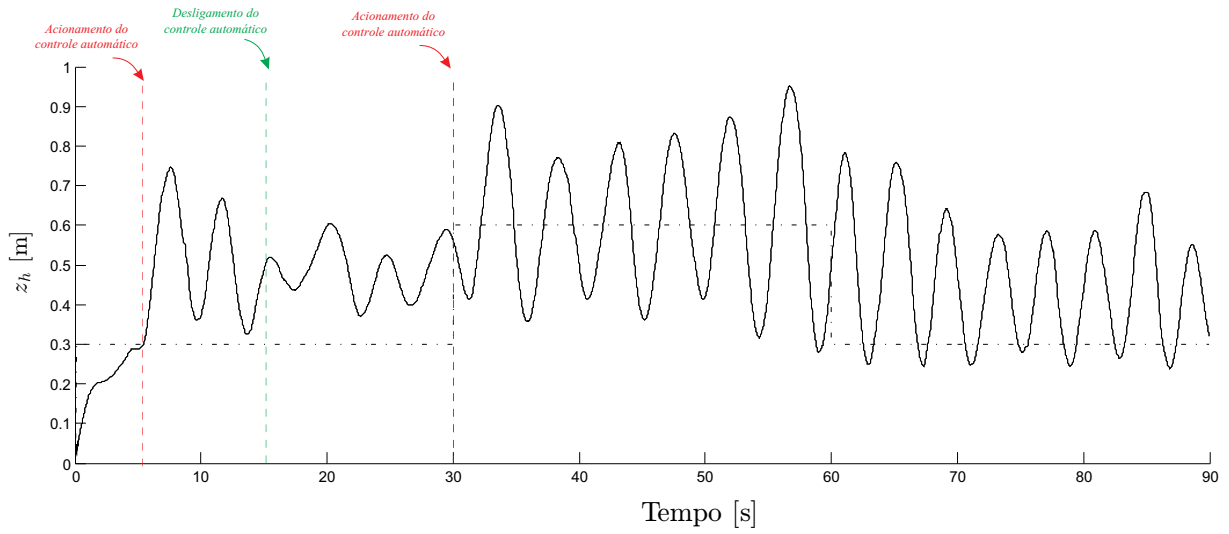


Figura 53: Controle autônomo da altura.

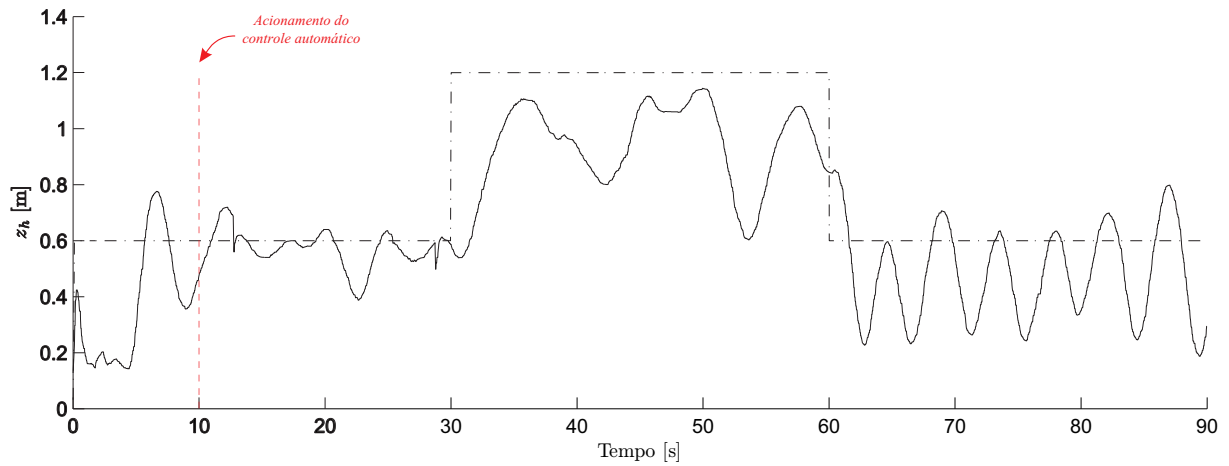


Figura 54: Controle autônomo da altura.

escolhidos foram de 60cm para o valor mais baixo e 120cm para o mais alto. O resultado deste experimento é apresentado na Figura 54. É possível visualizar que o controlador conseguiu estabilizar no valor inicial de 60cm, mas quando o valor desejado foi alterado para 120cm ele não conseguiu atingir, provando que os ganhos ajustados foram menores que o desejável. Por fim, ao voltar a referência para 60cm o veículo não consegue se estabilizar, tendo novamente a característica de um sistema instável.





## 6 Conclusões

*“Sempre confiando em quem somos,  
e nada mais importa”*

*(James Hetfield)*

O objetivo desta Dissertação, visa a criação de uma plataforma para suporte a voos autônomos ou remotamente controlados por rádio controle de helimodelos, contemplando tanto a parte de *hardware* quanto a de *software*. Seguindo este objetivo, foi projetada e confeccionada a placa *AuRoRa Board*, que tem a capacidade de ler todos os dados do rádio controle e interpretá-los, podendo atuar ou não sobre os servomotores do helimodelo de acordo com tais sinais. Ao mesmo tempo, tal placa está apta a receber os sinais de controle sintetizados por controladores de alto nível projetados para controlar a navegação dos helimodelos, executados em computadores embarcados no próprio veículo ou em computadores em terra. Então, conforme decisão do usuário, os sinais a serem enviados aos servomotores são selecionados, dentre os que provêm do rádio controle e dos controladores de alto nível, a placa é capaz de atuar em todos os servomotores do veículo. Esta atuação se dá através do envio dos sinais selecionados para um módulo de estabilização do veículo em voo pairado, denominado *Helicommand*, integrado à placa *AuRoRa Board*, que então atua nos servomotores.

Tanto a leitura dos sinais quanto a atuação nos servomotores é feita com uma resolução adequada, sendo que pequenas variações dos valores são imperceptíveis mesmo vistas de um osciloscópio. A placa *AuRoRa Board* também é responsável por toda comunicação com os controladores de alto nível, enviando as informações sobre o rádio controle e o nível das baterias e recebendo os comandos vindos do controlador e aplicando-os aos servomotres via *Helicommand*.

O módulo *Helicommand* foi inicialmente integrado aos helimodelos utilizados, o T-REX 450 e o T-REX 600, e configurado para os mesmos. Ele apresenta uma boa estabilização do veículo em voo pairado, porém apenas quando o controle, indiferente se o rádio controle ou o controle automático, se encontra em sua posição neutra. Como

funcionalidade adicional, é possível variar a amplitude dos comandos, para mais ou para menos, podendo, assim, limitar os movimentos do helimodelo.

Um dos canais de comunicação da placa *AuRoRa Board* é com um computador embarcado modelo *PicoITX*, instalado a bordo do helimodelo T-REX 600, para que os controladores automáticos de voo possam ser executados a bordo da própria aeronave.

A princípio foi realizada uma fusão sensorial dos sensores da unidade inercial da aeronave, apresentada no Apêndice B, a uma frequência de  $45Hz$ , a qual não acompanha os movimentos do helimodelo com fidelidade. Desta forma, foi utilizado uma frequência de filtragem de  $200Hz$ , com a captura dos dados a  $45Hz$ , que é a frequência de atualização dos dados do helimodelo. Assim, foi obtida uma confiabilidade nos dados filtrados.

Porém, ainda assim existe uma grande interferência por parte do magnetômetro. Estes sensores são influenciados principalmente pelo fluxo magnético provocado pelo motor do helimodelo, o qual gera um erro de orientação. O erro, após um certo nível de velocidade do rotor principal, é praticamente constante. Isto não provoca problemas a nível de controle, porém deixa de marcar o norte verdadeiro.

Em adição à placa projetada, e instalada a bordo da aeronave, também foi desenvolvido um *software*, a plataforma *AuRoRa*, para atuar integradamente com a placa *AuRoRa Board*, permitindo que os sinais gerados por controladores executados em um computador em terra sejam transmitidos, via um canal *Bluetooth*, um enlace *Zigbee*, ou mesmo uma rede *wireless* ou UDP, para atuarem no helimodelo. Este canal de comunicação também permite à plataforma receber os dados de voo (leituras dos sensores a bordo da aeronave) e armazená-los para análise posterior do voo, ou então exibi-los *online*, no mesmo computador que gera os sinais de controle ou em outro computador, conectado àquele via protocolo UDP.

Assim é que a plataforma implementada, no que se refere ao *hardware* e ao *software*, se mostrou adequada para a tarefa a que se destina, permitindo, inclusive, que vários veículos possam ser simulados ao mesmo tempo, ou mesmo que se realizem experimentos reais com vários veículos, servindo como suporte para sistemas que lidam com a coordenação de grupos de veículos aéreos ou mesmo veículos aéreos e terrestres.

Consequentemente, pode-se dizer que o objetivo original foi plenamente atingido, com a plataforma de *hardware* e de *software* descrita nesta Dissertação.

## *APÊNDICE A – Fusão Sensorial*

Para o controle autônomo de um robô é necessário fornecer informações sensoriais confiáveis ao mesmo, o caso do VANT não é diferente. Como já citado, usar os dados provindos do acelerômetro, giroscópio ou magnetômetro sem uma devida filtragem não irá fornecer os dados inerciais de velocidade e aceleração com a devida confiabilidade, por isso é necessário a realização de uma filtragem ou fusão sensorial.

Existem diversos trabalhos na literatura que utilizam de filtragens para a obtenção de ângulos de arfagem, rolagem e guinada confiáveis, além de uma eventual correção dos valores obtidos pelos sensores, pois os dados sensoriais, usualmente, são integrados para determinação de outras variáveis, logo, estes, então, necessitam de ter seus erros minimizados.

Na literatura existem diversas técnicas que tratam destes sensores a fim de entregar dados sensoriais confiáveis ao veículo. Muitas delas utilizam de filtragens utilizando Kalman,[71, 72, 73, 74]. Existem ainda trabalhos que utilizam filtros de Kalman juntamente com quatérnios [75]. Neste trabalho será utilizado a mesma abordagem adota em trabalhos anteriores do mesmo grupo, ou seja, a filtragem realizada em [76, 46].

### **A.1 Quatérnios**

Quando necessita-se de realizar uma modelagem ou controle de um veículo na robótica, é preciso, majoritariamente, a especificação de sua posição e orientação. Determinar a orientação de um objeto no espaço é realizar translações lineares em torno dos eixos ortogonais a partir de uma origem conhecida, logo, sua representação é um conjunto de variáveis cartesianas. Representar a orientação não é análogo a posição.

A dificuldade inerente da representação da orientação é devido a propriedade de comutação das operações de rotação. Ou seja, para translações a ordem de aplicação das translações não influencia no resultado final, pois são operações lineares e independente

uma das outras, já para o caso de rotações a ordem da realização das mesmas influencia no resultado final [77].

A utilização das matrizes de rotação e ângulos de Euler é a forma mais comum da apresentação de rotações no espaço, entretanto, é possível a representação por outra forma, pela utilização de quatérnios. Os quatérnios foram criados em 1843, pelo físico, astrônomo e matemático William Rowan Hamilton e designam um vetor de quatro dimensões, uma real e três imaginárias e este pode representar a orientação de um corpo rígido ou de eixos de coordenadas no espaço tridimensional [77].

## A.2 Operações por Quatérnios

Uma orientação de um referencial  $B$  relativo a um plano  $A$  pode ser realizado através de uma rotação de um ângulo  $\theta$  em torno do eixo  ${}^A\hat{r}$ , este, definido no plano  $A$ . A representação desta rotação é observada na Figura 55, enquanto o quatérnio que representa esta orientação,  ${}_B^A\hat{q}$ , é apresentado na Equação A.1, onde  $r_x$ ,  $r_y$  e  $r_z$  são os componentes  $x$ ,  $y$  e  $z$ , respectivamente, do vetor unitário  ${}^A\hat{r}$  do referencial  $A$ .

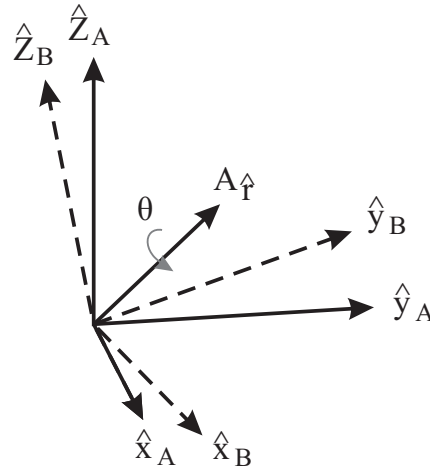


Figura 55: Orientação dos eixos coordenados de de B para A utilizando quatérnios.

$$q = [q_1 \ q_2 \ q_3 \ q_4] = \left[ \cos\left(\frac{\theta}{2}\right) \ r_x \cos\left(\frac{\theta}{2}\right) \ r_y \cos\left(\frac{\theta}{2}\right) \ r_z \cos\left(\frac{\theta}{2}\right) \right] \quad (\text{A.1})$$

Para facilitar o entendimento, neste trabalho foram adotadas as notações de [46], em que  ${}_B^A\hat{q}$  descreve a orientação do referencial  $B$  em relação ao  $A$  e  ${}^A\hat{r}$  é o vetor descrito no plano  $A$ .

O conjugado de um quatérnio é a representação da operação inversa, ou seja, o

conjugado de  ${}^A_B\hat{q}$  é  ${}^B_A\hat{q}$ . O conjugado de um quatérnio é representado pelo  $*$  e definido como

$${}^A_B\hat{q}^* = {}^B_A\hat{q} = [q_1 \ -q_2 \ -q_3 \ -q_4]. \quad (\text{A.2})$$

Sua soma é feita elemento a elemento, dois quatérnios  ${}^A_B\hat{q} = [q_1 \ -q_2 \ -q_3 \ -q_4]$  e  ${}^B_C\hat{q} = [q_6 \ -q_7 \ -q_8 \ -q_9]$  têm sua soma realizada da seguinte forma

$${}^A_B\hat{q} + {}^B_C\hat{q} = \begin{bmatrix} q_1 & + & q_6 \\ q_2 & + & q_7 \\ q_3 & + & q_8 \\ q_4 & + & q_9 \end{bmatrix}^T. \quad (\text{A.3})$$

O produto de quatérnios é detonado pelo símbolo  $\otimes$  e pode ser utilizado para definir sucessivas rotações. Partindo de duas orientações  ${}^A_B\hat{q}$  e  ${}^B_C\hat{q}$ , a orientação  ${}^A_C\hat{q}$  é definida por

$${}^A_C\hat{q} = {}^A_B\hat{q} \otimes {}^B_C\hat{q}. \quad (\text{A.4})$$

O produto de dois quatérnios  $a$  e  $b$  é definido como:

$$a \otimes b = [a_1 \ a_2 \ a_3 \ a_4] \otimes [b_1 \ b_2 \ b_3 \ b_4] = \begin{bmatrix} a_1b_1 - a_2b_2 - a_3b_3 - a_4b_4 \\ a_1b_2 - a_2b_1 - a_3b_4 - a_4b_3 \\ a_1b_3 - a_2b_4 - a_3b_1 - a_4b_2 \\ a_1b_4 - a_2b_3 - a_3b_2 - a_4b_1 \end{bmatrix}^T, \quad (\text{A.5})$$

atentando-se para o fato de que seu produto não é comutativo, ou seja,  $a \otimes b \neq b \otimes a$ .

A multiplicação pode ser utilizada para rotacionar um vetor tridimensional pela utilização de um quatérnio, sendo esta operação definida como

$${}^Bv = {}^A_B\hat{q} \otimes {}^Av \otimes {}^A_B\hat{q}^* \quad (\text{A.6})$$

${}^Av$  e  ${}^Bv$  são a representação de um mesmo vetor, porém, o primeiro descrito em relação ao referencial  $A$  enquanto o segundo ao  $B$ . Estes vetores possuem um 0 inserido como seu primeiro elemento os tornando um vetor de 4 elementos, validando assim, a operação descrita acima.

A determinação de uma determinada orientação  ${}^A_B\hat{q}$  pode ser representada pela matriz de rotação  ${}^A_BR$ , definida por,

$${}^A_BR = \begin{bmatrix} 2q_1^2 - 1 + 2q_2^2 & 2(q_1q_3 + q_1q_4) & 2(q_2q_4 - q_1q_3) \\ 2(q_1q_3 + q_1q_4) & 2q_1^2 - 1 + 2q_3^2 & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 + q_1q_3) & 2(q_3q_4 + q_1q_2) & 2q_1^2 - 1 + 2q_4^2 \end{bmatrix} \quad (\text{A.7})$$

Os ângulos de Euler  $\phi$ ,  $\theta$  e  $\psi$  descrevem a orientação do referencial  $B$  obtida por sucessivas rotações sobre o referencial  $A$ , de  $\phi$  sobre  $\hat{x}_B$ ,  $\theta$  sobre  $\hat{y}_B$  e  $\psi$  sobre  $\hat{z}_B$ . A representação por ângulos de Euler são definidas pelas Equações A.8, A.9 e A.10.

$$\phi = \text{Atan2}(2q_2q_3 - 2q_1q_4, 2q_1^2 + 2q_2^2 - 1), \quad (\text{A.8})$$

$$\theta = -\text{sen}^{-1}(2q_2q_4 + 2q_1q_3) \quad (\text{A.9})$$

$$\psi = \text{Atan2}(2q_3q_4 - 2q_1q_2, 2q_1^2 + 2q_4^2 - 1), \quad (\text{A.10})$$

### A.3 Filtragem por Quatérnios

As medidas do giroscópios são rotações respectivas a cada eixo, dadas em  $rad/s$ , desta forma, eles podem ser arrumados da seguinte forma:

$$s_\omega = [\omega_x \ \omega_y \ \omega_z], \quad (\text{A.11})$$

onde  $\omega_x$ ,  $\omega_y$  e  $\omega_z$  são as velocidades angulares em torno dos eixos subscritos. A derivada do quatérnio que descreve a taxa de variação da orientação do frame da terra em relação ao frame dos sensores,  ${}^S_E\dot{q}$ , pode ser calculado por meio de [78].

$${}^S_E\dot{q} = \frac{1}{2} {}^S_E\hat{q} \otimes {}^S\omega \quad (\text{A.12})$$

$${}^S_E\dot{q}_{\omega,t} = \frac{1}{2} {}^S_E\hat{q}_{est,t-1} \otimes {}^S\omega_t, \quad (\text{A.13})$$

$${}^S_E q_{\omega,t} = {}^S_E \hat{q}_{est,t-1} \otimes + {}^S_E \dot{q}_{\omega t} \Delta t, \quad (\text{A.14})$$

onde  $\Delta t$  é o tempo de amostragem e  ${}^S_E \hat{q}_{est,t-1}$  é a estimativa anterior da orientação. Esta é obtida com base nas três leituras de um giroscópio, porém, como já mencionado este sensor apresenta um problema de *drift*, além de erros de conversão analógica/digital, desta forma não deve ser utilizado sozinho para determinação de uma orientação.

## A.4 Orientação a Partir de Observação de Vetores

Um acelerômetro de três eixos mede a gravidade da Terra juntamente com as acelerações lineares aplicadas ao mesmo. Analogamente um magnetômetro de três eixos mede o campo gravitacional terrestre juntamente com suas perturbações. Conhecendo a orientação dos vetores oriundos destes sensores é possível calcular a orientação do referencial dos sensores em relação ao terrestre. Desta forma, irá ser adotado um problema de otimização, onde o quatérnio  ${}^S_E \hat{q}$  representa a orientação da placa dos sensores, que alinha a direção de referencia do campo conhecido no referencial da Terra,  ${}^E_d$ , com a direção do campo medido no referencial dos sensores,  ${}^S_s$ , através da rotação descrita em A.6, que irá rotacionar estes sensores através do quatérnio estimado para a orientação corrente[76]. Desta forma, ele,  ${}^S_E \hat{q}$ , é encontrado como solução da equação

$$\min f({}^S_E \hat{q}, {}^E_d, {}^S_s), \quad (\text{A.15})$$

onde  $f({}^S_E \hat{q}, {}^E_d, {}^S_s)$  é a função objetivo que caracteriza o erro de orientação, definido como

$$f({}^S_E \hat{q}, {}^E_d, {}^S_s) = {}^S_E \hat{q}^* \otimes {}^E_d \otimes {}^S_E \hat{q} - {}^S_s \quad (\text{A.16})$$

os vetores supracitados são mostrados nas Equações A.17, A.18 e A.19

$${}^S_E \hat{q} = [q_1 \ q_2 \ q_3 \ q_4] \quad (\text{A.17})$$

$${}^E_d = [0 \ d_x \ d_y \ d_z] \quad (\text{A.18})$$

$${}^S\hat{s} = [0 \ s_x \ s_y \ s_z] \quad (\text{A.19})$$

É proposto uma função para minimizar o custo da função objetivo, será adotado o método de gradiente descendente para tal, uma vez que é mais simples de implementar e mais rápido computacionalmente. A Equação A.20 descreve este algoritmo para  $n$  iterações, resultando em uma estimativa de orientação,  ${}^S\hat{q}_{n+1}$ , baseada em uma orientação inicial arbitrária  ${}^S\hat{q}_0$  e um passo de  $\mu$ . A Equação A.21 computa o gradiente descendente da superfície de solução definida pela função objetivo e seu jacobiano, simplificado para o vetor de três linhas definido pelas equações A.22 e A.23.

$${}^S\hat{q}_{k+1} = {}^S\hat{q}_k - \mu \frac{\delta f({}^S\hat{q}, {}^E\hat{d}, {}^S\hat{s})}{\|f({}^S\hat{q}, {}^E\hat{d}, {}^S\hat{s})\|}, \quad k = 1, 2, \dots, n \quad (\text{A.20})$$

$$\Delta f({}^S\hat{q}, {}^E\hat{d}, {}^S\hat{s}) = J^T({}^S\hat{q}_k, {}^E\hat{d}) f({}^S\hat{q}, {}^E\hat{d}, {}^S\hat{s}) \quad (\text{A.21})$$

$$f({}^S\hat{q}, {}^E\hat{d}, {}^S\hat{s}) = \begin{bmatrix} 2d_x(\frac{1}{2} - q_3^2 - q_4^2) + 2d_y(q_1q_4 + q_2q_3) + 2d_z(q_2q_4 - q_1q_3) - s_x \\ 2d_x(q_2q_3 + q_1q_4) + 2d_x(\frac{1}{2} - q_2^2 - q_4^2) + 2d_x(q_1q_2 - q_3q_4) - s_y \\ 2d_x(q_1q_3 + q_2q_4) + 2d_y(q_3q_4 - q_1q_2) + 2d_x(\frac{1}{2} - q_2^2 - q_3^2) - s_z \end{bmatrix} \quad (\text{A.22})$$

$$J({}^S\hat{q}_k, {}^E\hat{d}) = \begin{bmatrix} 2d_yq_4 - 2d_zq_3 & 2d_yq_3 + 2d_zq_4 & -4d_xq_3 + 2d_yq_2 - 2d_zq_1 & -4d_xq_4 + 2d_yq_1 + 2d_zq_2 \\ -2d_xq_4 - 2d_zq_2 & 2d_xq_3 - 4d_yq_2 + 2d_zq_1 & 2d_xq_2 + 2d_zq_4 & -2d_xq_1 - 4d_yq_4 + 2d_zq_3 \\ 2d_zq_3 - 2d_yq_2 & 2d_xq_4 - 2d_yq_1 - 4d_zq_2 & 2d_xq_1 + 2d_yq_4 - 4d_zq_3 & 2d_xq_2 + 2d_yq_3 \end{bmatrix} \quad (\text{A.23})$$

As Equações A.22 e A.23 descrevem, de forma geral, o algoritmo aplicável a um campo definido em qualquer direção. Entretanto, algumas simplificações podem ser realizadas nas equações acima. Irá ser definido um vetor que representa a gravidade da terra, ou seja,  ${}^Eg = {}^E\hat{d} = [0 \ 0 \ 0 \ 1]$ , tendo assim somente valor no componente  $z$ . Define-se também o vetor contendo as medidas normalizadas do acelerômetro  ${}^S\hat{a} = {}^S\hat{s} = [0 \ a_x \ a_y \ a_z]$ . Substituindo estas duas considerações nas equações A.22 e A.23, obtém-se:

$$f_g({}^S\hat{q}, {}^S\hat{a}) = \begin{bmatrix} 2(q_2q_3 - q_1q_3) - a_x \\ 2(q_1q_2 - q_3q_4) - a_y \\ 2(\frac{1}{2} - q_2^2 - q_3^2) - a_z \end{bmatrix} \quad (\text{A.24})$$



$$f_g({}^S_E\hat{q}, {}^S\hat{a}) = \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{bmatrix} \quad (\text{A.25})$$

onde o subscrito  $g$  indica que a função objetivo esta com relação ao acelerômetro.

O campo gravitacional terrestre pode ser considerado como um vetor de duas dimensões projetado sobre o eixo  $z$  do referencial. Desta forma ele será considerado como um vetor com componente em um dos dois eixos horizontais ou seja,  $x$  ou  $y$ , e um componente sobre o eixo vertical  $z$ . Este componente horizontal é devido a declinação do campo gravitacional, que, para o Brasil é de aproximadamente  $-37^\circ$  [ ]. Desta forma, o vetor referência é definido como  ${}^E\hat{d} = {}^E\hat{b} = [0 \ b_x \ 0 \ b_z]$  e o vetor contendo as medidas do magnetômetro normalizadas definido como  ${}^S\hat{m} = [m_x \ m_y \ m_z]$ . Fazendo a substituição destes vetores nas equações A.22 e A.23, é obtido as equações , onde o subscrito  $b$  indica que as mesmas foram obtidas pelo magnetômetro.

$$f_b({}^S_E\hat{q}, {}^E\hat{b}, {}^S\hat{m}) = \begin{bmatrix} 2b_x(0,5 - q_3^2 - q_4^2) + 2b_z(q_2q_4 - q_1q_3) - m_x \\ 2b_x(q_2q_3 - q_1q_4) + 2b_z(q_1q_2 - q_3q_4) - m_y \\ 2b_x(q_1q_3 - q_2q_4) + 2b_z(0,5 - q_2^2 - q_3^2) - m_z \end{bmatrix} \quad (\text{A.26})$$

$$f_g({}^S_E\hat{q}, {}^S\hat{a}) = \begin{bmatrix} -2b_zq_3 & 2b_zq_4 & -4b_xq_3 - 2b_zq_1 & -4b_xq_4 + 2b_zq_2 \\ -2b_xq_4 + 2b_zq_2 & 2b_xq_3 + 2b_zq_1 & 2b_xq_2 + 2b_zq_4 & -2b_xq_1 + 2b_zq_3 \\ 2b_xq_3 & 2b_xq_4 - 4b_zq_2 & 2b_xq_1 - 4b_zq_3 & 2b_xq_2 \end{bmatrix} \quad (\text{A.27})$$

A fim de obter um resultado único, é feita a combinação das duas equações de referência (acelerômetro e magnetômetro), desta forma,

$$f_{g,b}({}^S_E\hat{q}, {}^S\hat{a}, {}^E\hat{b}, {}^S\hat{m}) = \begin{bmatrix} f_g({}^S_E\hat{q}, {}^S\hat{a}) \\ f_b({}^S_E\hat{q}, {}^E\hat{b}, {}^S\hat{m}) \end{bmatrix} \quad (\text{A.28})$$

$$J_{g,b}({}^S_E\hat{q}, {}^E\hat{b}) = \begin{bmatrix} J_g^T({}^S_E\hat{q}) \\ J_b^T({}^S_E\hat{q}, {}^E\hat{b}) \end{bmatrix} \quad (\text{A.29})$$

Como resultado, tem-se a Equação A.30, que calcula a orientação estimada  ${}^S_E$  em um determinado instante  $t$  baseado em uma orientação anterior  ${}^E$  e no gradiente da função objetivo,  $\delta f$ , que é definido pelas medidas dos sensores  ${}^S$  e  ${}^S$  no mesmo instante de tempo.

$${}^S_E q_{\nabla,t} = {}^S_E \hat{q}_{est,t-1} - \mu_t \frac{\nabla f}{\|\nabla f\|} \quad (\text{A.30})$$

É característica de algoritmos de otimização a necessidade de múltiplas iterações para a obtenção da solução otimizada. Para este método, descrito na equação A.30, é aceitável sua realização a cada iteração. Isto será válido desde que a taxa de aprendizado  $\mu_t$  seja igual ou maior a taxa física de variação de orientação. Ou seja, como é conhecido a estrutura dos sensores, pode-se escolher um valor para o passo do filtro o qual o mesmo convergirá igual ou mais rápido que a variação sensorial, que neste caso será utilizado a velocidade angular do giroscópio, isto garante que a taxa de convergência  ${}^S_E q_{\delta,t}$  seja limitada por esta variação. A Equação A.31 apresenta tal restrição, onde  ${}^S_E \dot{q}_{\omega,t}$  é a taxa física de orientação, medida pelo giroscópio e  $\alpha$  um complemento de  $\mu$ , para considerar ruídos presentes nas medidas dos acelerômetros e magnetômetros.

$$\mu_t = \alpha \|{}^S_E \dot{q}_{\omega,t}\| \Delta t, \alpha > 1 \quad (\text{A.31})$$

#### A.4.1 Algoritmo de Fusão

A orientação estimada do referencial dos sensores relativo ao referencial terrestre,  ${}^S_E q_{est,t}$ , é obtido através da fusão das orientações calculadas por  ${}^S_E q_{\omega,t}$  e  ${}^S_E q_{\nabla,t}$  pelas equações A.14 e A.30 respectivamente. Esta fusão pode ser realizada da seguinte forma:

$${}^S_E q_{est,t} = \gamma_t {}^S_E q_{\nabla,t} + (1 - \gamma_t) {}^S_E q_{\omega,t}, \quad 0 \leq \gamma_t \leq 1, \quad (\text{A.32})$$

onde  $\gamma_t$  e  $(1 - \gamma_t)$  são os pesos aplicados a cada orientação.

Um valor ótimo para  $\gamma_t$  pode ser definido como aquele que assegure que divergência de  ${}^S_E q_{\omega}$  será igual a convergência de  ${}^S_E q_{\nabla}$ . Esta relação está mostrada na Equação A.33 onde  $\frac{\mu_t}{\Delta t}$  é a taxa de convergência de  ${}^S_E q_{\nabla}$  e  $\beta$  a taxa de divergência de  ${}^S_E q_{\omega}$ . Ambos expressados em função do erro de medida do giroscópio.

$$(1 - \gamma_t)\beta = \gamma_t \frac{\mu_t}{\Delta t} \quad (\text{A.33})$$

Rearranjando a Equação A.34 é obtido a Equação A.34.

$$\gamma_t = \frac{\beta}{\frac{\mu_t}{\Delta t} + \beta} \quad (\text{A.34})$$

Equações A.32 e A.34, asseguram a fusão ótima de  ${}^S_E q_{\omega,t}$  e  ${}^S_E q_{\nabla,t}$ , assumindo que taxa de convergência de  ${}^S_E q_{\nabla}$ , governada pela constante  $\alpha$ , é igual ou maior que a taxa física de variação dos valores medidos pelo giroscópios. Desta forma,  $\alpha$  não possui limite superior, sendo assim, pode-se assumi-lo como um valor muito grande e, devido à relação A.31,  $\mu_t$  também pode ter um valor grande, ornando o termo  ${}^S_E \hat{q}_{est,t-1}$  da Equação A.30 desprezível, resultando em

$${}^S_E q_{\nabla,t} \approx -\mu_t \frac{\nabla f}{\|\nabla f\|}. \quad (\text{A.35})$$

Da mesma forma, a Equação A.34 leva a

$$\gamma_t \approx \frac{\beta \Delta t}{\mu_t}, \quad (\text{A.36})$$

permitindo concluir que  $\gamma_t \approx 0$ .

Substituindo as equações A.14, A.35 e A.36 em A.32, obtém-se

$${}^S_E q_{est,t} = \frac{\beta \Delta t}{\mu_t} \left( -\mu_t \frac{\nabla f}{\|\nabla f\|} \right) + (1 - 0) \left( {}^S_E \hat{q}_{est,t-1} + {}^S_E \dot{q}_{\omega,t} \Delta t \right). \quad (\text{A.37})$$

$\gamma_t$  foi substituído, no primeiro termo, conforme a equação A.36, e no segundo termo como 0.

Pode ser realizada a simplificação da Equação A.37 levando a A.38, onde  ${}^S_E \dot{q}_{est,t}$  é a taxa de variação estimada da orientação, definida por A.39, e  ${}^S_E \hat{q}_{\epsilon,t}$  é a direção do erro de  ${}^S_E \dot{q}_{est,t}$ , definido por A.40.

$${}^S_E q_{est,t} = {}^S_E \hat{q}_{est,t-1} + {}^S_E \dot{q}_{est,t} \Delta t \quad (\text{A.38})$$

$${}^S_E \dot{q}_{est,t} = {}^S_E \dot{q}_{\omega,t} - \beta {}^S_E \hat{q}_{\epsilon,t} \quad (\text{A.39})$$

$${}^S_E \hat{q}_{\epsilon,t} = \frac{\nabla f}{\|\nabla f\|} \quad (\text{A.40})$$

Pode ser observado, das Equações A.38, A.39 e A.40, que o filtro calcula a orientação  ${}^S_E q_{est}$  pela integração numérica da variação de orientação  ${}^S_E \dot{q}_{est}$ , este é, por sua vez, calculado como a variação da orientação medida pelos giroscópios,  ${}^S_E \dot{q}_\omega$ , utilizando  $\beta$  para retirar os possíveis erros inerentes do sensor na direção do erro estimado  ${}^S_E \dot{q}_{\epsilon,t}$ , este originado pelas leituras do acelerômetro e magnetômetro.

### A.4.2 Compensação da Distorção Magnética

Magnetômetros são sensores que possuem grande interferência externas causadas pelo meio. Estas podem ser por quaisquer materiais ferromagnético ou magnéticos nas proximidades do sensor. Uma melhor abordagem das interferências deste sensor, juntamente com sua calibração pode ser visualizado no Apêndice .

Um erro comum que estes sensores apresenta é o de inclinação, ou seja, as leituras do sensor é afetada com a inclinação do mesmo [79]. Esta inclinação deve ser compensada a fim de evitar erros na identificação do norte geográfico.

Em um determinado instante de tempo  $t$ , a medida de direção do campo magnético no referencial terrestre,  ${}^E \hat{h}_t$ , pode ser definido como a direção do campo medido no plano dos sensores,  ${}^S \hat{m}_t$ , rotacionada pelo quatérnio  ${}^S_E \hat{q}_{est,t-1}$ , que é estimado pelo filtro de fusão, descrito na Equação A.41.

Como já citado, para contornar o efeito da medida do campo magnético com uma inclinação errada, pode-se tomar a medida de direção deste campo com mesma inclinação que o filtro. Ou seja, calcula-se  ${}^E \hat{b}_t$  como  ${}^E \hat{h}_t$ , tendo suas medidas normalizadas e com componentes apenas em  $x$  e  $z$ , conforme Equação A.42 [80]. Desta forma elimina-se a necessidade de estabelecer uma direção de referência fixa do campo magnético para o local em que o magnetômetro captura seus dados.

$${}^E \hat{h}_t = \begin{bmatrix} 0 & h_x & h_y & h_z \end{bmatrix} = {}^S_E \hat{q}_{est,t-1} \otimes {}^S \hat{m}_t \otimes {}^S_E \hat{q}_{est,t-1}^* \quad (\text{A.41})$$

$${}^E \hat{b}_t = \begin{bmatrix} 0 & \sqrt{h_x^2 + h_y^2} & 0 & h_z \end{bmatrix} \quad (\text{A.42})$$

### A.4.3 Compensação do *Drift* do Giroscópio

Como já apresentado, giroscópios são comuns por apresentar um deslize na medida a partir da sua calibração de zero inicial, denominado *drift*. Esta característica deve ser

contemplada na criação de filtros. Utilizando o método de Kalman, ele pode ser modelado como um estado adicional no modelo [81, 82]. Uma forma mais simples da retirada dessa *bias*, pode ser através da integral do erro na taxa de variação da orientação [83].

Desta forma, em [80] é proposto que a direção normalizada do erro na taxa de variação da orientação,  ${}^S_E\dot{\hat{q}}_{\epsilon,t}$ , pode ser usada para expressar o erro de velocidade angular em cada eixo do giroscópio invertendo a equação A.13, resultando na equação A.43. A *bias* do giroscópio,  ${}^S\omega_{b,t}$ , é representado como a componente DC de  ${}^S\omega_{\epsilon,t}$  e, como tal, deve ser removido como a integral de  ${}^S\omega_{\epsilon,t}$  associada a um ganho  $\zeta$ . Isto implicará em uma velocidade angular compensada  ${}^S\omega_{c,t}$ , conforme visto nas equações A.44 e A.45 [76].

$${}^S\omega_{\epsilon,t} = 2{}^S_E\hat{q}_{est,t-1}^* \otimes {}^S_E\dot{\hat{q}}_{\epsilon,t} \quad (\text{A.43})$$

$${}^S\omega_{b,t} = \zeta \sum_t {}^S\omega_{\epsilon,t} \Delta t \quad (\text{A.44})$$

$${}^S\omega_{c,t} = {}^S\omega_t - {}^S\omega_{b,t} \quad (\text{A.45})$$



## ***APÊNDICE B – AuRoRa Board***

Com o intuito de aumentar o entendimento sobre o funcionamento do *AuRoRa Board*, são apresentado neste apêndice as rotinas de inicialização e utilização dos periféricos do microcontrolador. Ainda é apresentado as camadas do circuito juntamente com o esquema elétrico da placa.

### **B.1 Códigos da Plataforma *AuRoRa* Comuns aos Helimodelos TREX-450 e TREX-600**

A seguir serão apresentados os códigos presentes na *AuRoRa Board*. Como a placa é utilizada para dois veículos distintos, algumas rotinas para o T-REX 450 são diferentes daquelas similares para o T-REX 600. Seja por configuração física das saídas e posicionamento das mesmas no helimodelo ou até mesmo pela diferença entre os módulos utilizados. Desta forma, serão mostrado, primeiramente, as rotinas comuns aos dois veículos, apresentadas abaixo. Vale ressaltar que cada seção está nomeada de acordo com o nome da rotina.

Como já citado, o Rádio Controle DX8, presente no T-REX 600, possui dois canais de comando, os quais possuem 3 estados, enquanto o DX7 apresenta dois canais de comando, os quais possuem apenas 2 estados.

#### **B.1.1 main.c**

```
#include "p24fj128gb206.h"
#include "delay.h"
#include "main.h"
_CONFIG1(ICS_PGx2 & FWDTEN_OFF & JTAGEN_OFF)
_CONFIG2( POSCMOD_HS & FCKSM_CSECME & FNOSC_PRI & PLLDIV_NODIV)
extern unsigned int ic1_period, ic2_period, ic3_period, ic4_period, ic5_period, ic6_period, ic7_period;
void IOInit_450();
void IOInit_600();
void ICInit();
```

```

void ADCInit();
void PWM_init();
void UART1Init();
void UART2Init();
void teste_ic();
int i = 0;

/*****
/*                                     MAIN                                     */
*****/
int main (void)
{
    TRISEbits.TRISE4 = 0;    //LED Vermelho
    pisca_led_verm(10);

/*****          INIT TIMERS          *****/
    //***** Timer 2 *****/
    T2CONbits.TON = 0;
    T2CONbits.TSIDL = 0;
    T2CONbits.TCKPS = 0x1; // Timerx Input Clock Prescale Select bits - 01= 1:8
    IPC1bits.T2IP = 0b111; // Seta o nível de prioridade da interrupção do Timer2
    IEC0bits.T2IE = 0;     // Habilita interrupção do timer2
    TMR2 = 0x0000;
    PR2 = 0xFFFF0;        // Carrega o valor do período 0xFFFF0
    T2CONbits.TON = 0;

    //***** Timer 3 *****/
    T3CONbits.TON = 0;     // Desabilita o Timer3
    IFS0bits.T3IF = 0;     // Limpa o flag de interrupção do timer3
    T3CONbits.TCKPS = 0b01; // Seleciona 1: Prescaler
    IPC2bits.T3IP = 0b010; // Seta o nível de prioridade da interrupção do Timer3
    IEC0bits.T3IE = 1;     // Habilita interrupção do timer3
    TMR3 = 0x0000;        // Limpa registrador do timer
    PR3 = 0x7500;         // Carrega o valor do período
    T3CONbits.TON = 0;

    pisca_led_verm(1);

    IOInit_450();
//    IOInit_600();
    ICInit();

    T2CONbits.TON = 1;        // Start Timer2
    pisca_led_verm(1);
    PORTBbits.RB14=0;
    teste_ic();

    //ADCInit();
    PWM_init();
    UART1Init();

    T3CONbits.TON = 1;        // Start Timer3

/*****

```



```
/*                                While true                                */
/*****/

    while(1)
    {
        __delay_ms(490);
        PORTCbits.RC14 = 1;
        __delay_ms(10);
        PORTCbits.RC14 = 0;

        __delay_ms(499);
        PORTCbits.RC14 = 1;
        __delay_ms(1);
        PORTCbits.RC14 = 0;

    }// while(1)
}

void teste_ic()
{
    if ((ic1_period > 3000) || (ic2_period > 3000) || (ic3_period > 3000)|| (ic4_period > 3000))
    {
        PORTEbits.RE3 = 1;
        PORTEbits.RE4 = 1;
        PORTCbits.RC14 = 1;
        __delay_s(1);
        asm("reset");
    }
}
```

### B.1.2 Init\_PWM.c

```
#include "p24fj128gb206.h"

void PWM_init(void)
{
    // Throttle - Baixo (1660) - Alto (2840)
    OC1CON1          = 0;
    OC1CON2          = 0;
    OC1R             = 1660;
    OC1RS            = 0x8200;
    OC1CON2bits.SYNCSEL = 0x1;
    OC1CON1bits.OCTSEL = 0x0;
    OC1CON1bits.OCFLT  = 0;
    OC1CON2bits.OCINV  = 0;
    OC1CON1bits.OCM    = 0b110;

    // Pra direita(1660) - Esquerda (2840)
    OC2CON1          = 0;
    OC2CON2          = 0;
    OC2R             = 2240;
    OC2CON2bits.SYNCSEL = 0x1;
    OC2CON1bits.OCTSEL = 0x0;
    OC2CON1bits.OCFLT  = 0;
```

```

OC2CON2bits.OCINV      = 0;
OC2CON1bits.OCM        = 0b110;

// Para Trás (1660) – Frente (2840)
OC3CON1                = 0;
OC3CON2                = 0;
OC3R                   = 2240;
OC3CON2bits.SYNCSEL    = 0x1;
OC3CON1bits.OCTSEL     = 0x0;
OC3CON1bits.OCFLT      = 0;
OC3CON2bits.OCINV      = 0;
OC3CON1bits.OCM        = 0b110;

// Tail – Direita (1660) – Esquerda (2840)
OC4CON1                = 0;
OC4CON2                = 0;
OC4R                   = 2240;
OC4CON2bits.SYNCSEL    = 0x1;
OC4CON1bits.OCTSEL     = 0x0;
OC4CON1bits.OCFLT      = 0;
OC4CON2bits.OCINV      = 0;
OC4CON1bits.OCM        = 0b110;

// Helicommand 1660–Laranja 2240–Vermelho 2840–Verde
OC5CON1                = 0;
OC5CON2                = 0;
OC5R                   = 1660;
OC5CON2bits.SYNCSEL    = 0x1;
OC5CON1bits.OCTSEL     = 0x0;
OC5CON1bits.OCFLT      = 0;
OC5CON2bits.OCINV      = 0;
OC5CON1bits.OCM        = 0b110;

// Elevator – Baixo (1660) – Alto (2840)
OC6CON1                = 0;
OC6CON2                = 0;
OC6R                   = 1660;
OC6CON2bits.SYNCSEL    = 0x1;
OC6CON1bits.OCTSEL     = 0x0;
OC6CON1bits.OCFLT      = 0;
OC6CON2bits.OCINV      = 0;
OC6CON1bits.OCM        = 0b110;

// Gyro Type
OC7CON1                = 0;
OC7CON2                = 0;
OC7R                   = 1660;
OC7CON2bits.SYNCSEL    = 0x1;
OC7CON1bits.OCTSEL     = 0x0;
OC7CON1bits.OCFLT      = 0;
OC7CON2bits.OCINV      = 0;
OC7CON1bits.OCM        = 0b110;
}

```

### B.1.3 Init\_ADC.c

```
#include "p24fj128gb206.h"

unsigned int    ULT1,ULT2,BAT1,BAT2,BAT3;
void ADC_Init(void);
void __attribute__((interrupt,no_auto_psv)) _ADC1Interrupt(void);

void ADCInit(void)
{
  ANCFG=0;
  ANSB = 0x0C07;
  AD1CSSL = 0x0C07;
  AD1CON1bits.ADSIDL = 1;
  AD1CON1bits.FORM = 0;
  AD1CON1bits.SSRC = 0b111;
  AD1CON1bits.ASAM = 1;
  AD1CON1bits.SAMP = 1;
  AD1CON2bits.VCFG = 0;
  AD1CON2bits.CSCNA = 1;
  AD1CON2bits.SMPI = 4;
  AD1CON2bits.ALTS = 0;
  AD1CON3bits.SAMC = 0b11111;
  AD1CON3bits.ADCS = 0xFF;
  AD1CON3bits.ADRC = 0;
  AD1CHSbits.CH0SB = 3;
  AD1CHSbits.CH0SA = 3;
  IEC0bits.AD1IE = 1;
  IPC3bits.AD1IP = 3;
  IFS0bits.AD1IF = 0;
  AD1CON1bits.ADON = 1;
}
void __attribute__((interrupt,no_auto_psv)) _ADC1Interrupt(void)
{
  IFS0bits.AD1IF = 0;
  BAT1 = ADC1BUF0*1.04;
  BAT2 = ADC1BUF1*1.85;
  BAT3 = ADC1BUF2*3.55;
  ULT1 = ADC1BUF3;
  ULT2 = ADC1BUF4;
}
```

### B.1.4 Init\_UART2.c

```
#include "p24fj128gb206.h"

unsigned char    buffer_u2[49],
                Char_Lat_D[4],Char_Lon_D[3],
                Char_Lat_m[6],Char_Lon_m[6],
                Time_c[7];

int Latitude_D,Longitude_D;
long int Latitude_m,Longitude_m,Time;
unsigned int     n2;
```

```

void UART2Init() {
U2BRG = 12;      // 24Mhz osc , U1BRG = 12 = 57600 Baud
U2MODE =        0x0000; //turn on module0x8000;
U2STA  =        0x8480; //set interrupts

IPC7bits.U2RXIP0 = 1;
IEC1bits.U2RXIE = 0; //Habilita interrupcao RX1
IFS1bits.U2RXIF = 0; //reset RX interrupt flag
}
void putcU2(char c){
if (!U2STAbits.UTXBF)
    U2TXREG = c;
while (!U2STAbits.TRMT);}
unsigned char getcU2(void){
U2STAbits.URXDA=0;
return U2RXREG;}
void __attribute__((interrupt , no_auto_psv)) _U2RXInterrupt(void) {
IFS1bits.U2RXIF = 0;
buffer_u2[n2]=getcU2();
n2++;
if (buffer_u2[0]!='$' || buffer_u2[n2-1]==' ' || buffer_u2[n2-1]==0x0A || buffer_u2[n2-1]==0x0D){
n2=0;}
else
{
if (n2==2 && buffer_u2[1]=='P'){
n2=0;}
if (n2==49){
n2=0;
if (buffer_u2[43]=='A')
{
Char_Lat_D[0]=buffer_u2[7];
Char_Lat_D[1]=buffer_u2[8];
Char_Lat_D[2]='\0';
Char_Lat_m[0]=buffer_u2[9];
Char_Lat_m[1]=buffer_u2[10];
Char_Lat_m[2]=buffer_u2[12];
Char_Lat_m[3]=buffer_u2[13];
Char_Lat_m[4]=buffer_u2[14];
Char_Lat_m[5]=buffer_u2[15];
Char_Lat_m[6]='\0';
Latitude_D = atoi(Char_Lat_D);
Latitude_m = atol(Char_Lat_m);
Char_Lon_D[0]=buffer_u2[19];
Char_Lon_D[0]=buffer_u2[20];
Char_Lon_D[1]=buffer_u2[21];
Char_Lon_D[2]='\0';
Char_Lon_m[0]=buffer_u2[22];
Char_Lon_m[1]=buffer_u2[23];
Char_Lon_m[2]=buffer_u2[25];
Char_Lon_m[3]=buffer_u2[26];
Char_Lon_m[4]=buffer_u2[27];
Char_Lon_m[5]=buffer_u2[28];
Char_Lon_m[6]='\0';
Longitude_D = atoi(Char_Lon_D);

```

```

Longitude_m = atol(Char_Lon_m);
Time_c[0]=buffer_u2[32];
Time_c[1]=buffer_u2[33];
Time_c[2]=buffer_u2[34];
Time_c[3]=buffer_u2[35];
Time_c[4]=buffer_u2[36];
Time_c[5]=buffer_u2[37];
Time_c[6]=buffer_u2[39];
Time_c[7]='\0';
Time = atol(Time_c);

putcU1('\n');
putcU1('$');
putcU1((Latitude_D&0xFF));
putcU1(((Latitude_m)&0xFF));
putcU1(((Latitude_m>>8)&0xFF));
putcU1(((Latitude_m>>16)&0xFF));
putcU1(((Latitude_m>>24)&0xFF));
putcU1((Longitude_D&0xFF));
putcU1(((Longitude_m)&0xFF));
putcU1(((Longitude_m>>8)&0xFF));
putcU1(((Longitude_m>>16)&0xFF));
putcU1(((Longitude_m>>24)&0xFF));
putcU1(((Time)&0xFF));
putcU1(((Time>>8)&0xFF));
putcU1(((Time>>16)&0xFF));
putcU1(((Time>>24)&0xFF));
}}}}

```

### B.1.5 Init\_IC

```
#include "p24fj128gb206.h"
```

```

unsigned int ic1_flag=0,ic1_int_counter=0,ic1_couter1, ic1_couter2,ic1_period;
unsigned int ic2_flag=0,ic2_int_counter=0,ic2_couter1, ic2_couter2,ic2_period;
unsigned int ic3_flag=0,ic3_int_counter=0,ic3_couter1, ic3_couter2,ic3_period;
unsigned int ic4_flag=0,ic4_int_counter=0,ic4_couter1, ic4_couter2,ic4_period;
unsigned int ic5_flag=0,ic5_int_counter=0,ic5_couter1, ic5_couter2,ic5_period;
unsigned int ic6_flag=0,ic6_int_counter=0,ic6_couter1, ic6_couter2,ic6_period;
unsigned int ic7_flag=0,ic7_int_counter=0,ic7_couter1, ic7_couter2,ic7_period;

int status_manual;
void ICInit(){
/*****
/*          Inicia Módulo de Captura    IC1          */
*****/

IC1CON1bits.ICI= 0b00;
IC1CON1bits.ICM= 0b001;
IC1CON1bits.ICTSEL = 0b001;
IC1CON2bits.SYNCSEL = 0b01100;
IEC0bits.IC1IE = 1;
IC2CON1bits.ICI= 0b00;
IC2CON1bits.ICM= 0b001;

```

```

IC2CON1bits.ICTSEL = 0b001;
IC2CON2bits.SYNCSEL = 0b10100;
IEC0bits.IC2IE = 1;
IC3CON1bits.ICI= 0b00;
IC3CON1bits.ICM= 0b001;
IC3CON1bits.ICTSEL = 0b001;
IC3CON2bits.SYNCSEL = 0b10100;
IEC2bits.IC3IE = 1;
IC4CON1bits.ICI= 0b00;
IC4CON1bits.ICM= 0b001;
IC4CON1bits.ICTSEL = 0b001;
IC4CON2bits.SYNCSEL = 0b10100;
IEC2bits.IC4IE = 1;
IC5CON1bits.ICI= 0b00;
IC5CON1bits.ICM= 0b001;
IC5CON1bits.ICTSEL = 0b001;
IC5CON2bits.SYNCSEL = 0b10100;
IEC2bits.IC5IE = 1;
IC6CON1bits.ICI= 0b00;
IC6CON1bits.ICM= 0b001;
IC6CON1bits.ICTSEL = 0b001;
IC6CON2bits.SYNCSEL = 0b10100;
IEC2bits.IC6IE = 1;
IC7CON1bits.ICI= 0b00;
IC7CON1bits.ICM= 0b001;
IC7CON1bits.ICTSEL = 0b001;
IC7CON2bits.SYNCSEL = 0b10100;
IEC1bits.IC7IE = 1;
}

/*****
/*          Interrupções de Captura          */
*****/

/*****          Input Capture 1          *****/
void __attribute__((interrupt,no_auto_psv)) _IC1Interrupt(void)
{
    ic1_int_counter++;
    if(ic1_int_counter == 1)
        ic1_couter1 = IC1BUF;
    else if(ic1_int_counter == 2)
    {
        ic1_couter2 = IC1BUF;
        ic1_flag = 1;
        if(ic1_flag);
        {
            if(ic1_couter2 > ic1_couter1)
                ic1_period = ic1_couter2 - ic1_couter1;
            else
                ic1_period = (PR2 - ic1_couter1) + ic1_couter2;
        }
        //if(status_manual == 0 || status_manual == 1 || status_manual == 2)
        {
            // Ganho Gyro
            OC7R = ic1_period;
        }
    }
}

```

```
}
ic1_flag = 0;
ic1_int_counter = 0;
}
}
IFS0bits.IC1IF=0;
}
/*****          Input Capture 2          *****/
void __attribute__((interrupt,no_auto_psv)) _IC2Interrupt(void)
{
ic2_int_counter++;
if(ic2_int_counter == 1)
ic2_couter1 = IC2BUF;
else if(ic2_int_counter == 2)
{
ic2_couter2 = IC2BUF;
ic2_flag = 1;
if(ic2_flag);
{
if(ic2_couter2 > ic2_couter1)
ic2_period = ic2_couter2 - ic2_couter1;
else
ic2_period = (PR2 - ic2_couter1) + ic2_couter2;
if(status_manual == 2)
{
// Elevator - Baixo (1660) - Alto (2840)
OC6R = ic2_period;
}
ic2_flag = 0;
ic2_int_counter = 0;
}
}
IFS0bits.IC2IF = 0;
}
/*****          Input Capture 3          *****/
void __attribute__((interrupt,no_auto_psv)) _IC3Interrupt(void)
{
ic3_int_counter++;
if(ic3_int_counter == 1)
ic3_couter1 = IC3BUF;
else if(ic3_int_counter == 2)
{
ic3_couter2 = IC3BUF;
ic3_flag = 1;
if(ic3_flag);
{
if(ic3_couter2 > ic3_couter1)
ic3_period = ic3_couter2 - ic3_couter1;
else
ic3_period = (PR2 - ic3_couter1) + ic3_couter2;
if (ic3_period > 2500 && ic3_period < 3000 )    //
{
PORTEbits.RE3 = 1;
status_manual = 1;
}
```

```

}
else if (ic3_period > 2000 && ic3_period < 2500)
{
PORTEbits.RE3 = 0;
status_manual = 0;
}
else if (ic3_period > 1500 && ic3_period < 2000)
{
PORTEbits.RE3 = !PORTEbits.RE3;
status_manual = -1;
}
ic3_flag = 0;
ic3_int_counter = 0;
}
}
IFS2bits.IC3IF = 0;
}
/*****          Input Capture 4          *****/
void __attribute__((interrupt,no_auto_psv)) _IC4Interrupt(void)
{
ic4_int_counter++;
if(ic4_int_counter == 1)
ic4_couter1 = IC4BUF;
else if(ic4_int_counter == 2)
{
ic4_couter2 = IC4BUF;
ic4_flag = 1;
if(ic4_flag);
{
if(ic4_couter2 > ic4_couter1)
ic4_period = ic4_couter2 - ic4_couter1;
else
ic4_period = (PR2 - ic4_couter1) + ic4_couter2;
if(status_manual == 2)
{
// Tail – Direita (1660) – Esquerda (2840)
OC4R = ic4_period;
}
ic4_flag = 0;
ic4_int_counter = 0;
}
}
IFS2bits.IC4IF = 0;
}
/*****          Input Capture 5          *****/
void __attribute__((interrupt,no_auto_psv)) _IC5Interrupt(void)
{
ic5_int_counter++;
if(ic5_int_counter == 1)
ic5_couter1 = IC5BUF;
else if(ic5_int_counter == 2)
{
ic5_couter2 = IC5BUF;
ic5_flag = 1;

```



```
if(ic5_flag);
{
if(ic5_couter2 > ic5_couter1)
ic5_period = ic5_couter2 - ic5_couter1;
else
ic5_period = (PR2 - ic5_couter1) + ic5_couter2;
if(status_manual == 1 || status_manual == 2)
{
// Para Trás (1660) - Frente (2840)
OC3R = ic5_period;
}
ic5_flag = 0;
ic5_int_counter = 0;
}
}
IFS2bits.IC5IF = 0;
}
/*****          Input Capture 6          *****/
void __attribute__((interrupt,no_auto_psv)) _IC6Interrupt(void)
{
ic6_int_counter++;
if(ic6_int_counter == 1)
ic6_couter1 = IC6BUF;
else if(ic6_int_counter == 2)
{
ic6_couter2 = IC6BUF;
ic6_flag = 1;
if(ic6_flag);
{
if(ic6_couter2 > ic6_couter1)
ic6_period = ic6_couter2 - ic6_couter1;
else
ic6_period = (PR2 - ic6_couter1) + ic6_couter2;
if(status_manual == 1 || status_manual == 2)
{
// Pra direita(1660) - Esquerda (2840)
OC2R = ic6_period;
}
ic6_flag = 0;
ic6_int_counter = 0;
}
}
IFS2bits.IC6IF = 0;
}
/*****          Input Capture 7          *****/
void __attribute__((interrupt,no_auto_psv)) _IC7Interrupt(void)
{
ic7_int_counter++;
if(ic7_int_counter == 1)
ic7_couter1 = IC7BUF;
else if(ic7_int_counter == 2)
{
ic7_couter2 = IC7BUF;
ic7_flag = 1;
```

```

if(ic7_flag);
{
if(ic7_couter2 > ic7_couter1)
ic7_period = ic7_couter2 - ic7_couter1;
else
ic7_period = (PR2 - ic7_couter1) + ic7_couter2;
// if(status_manual == 0 || status_manual == 1 || status_manual == 2)
{
// Throttle – Baixo (1660) – Alto (2840)
OCCR = ic7_period;
}
ic7_flag = 0;
ic7_int_counter = 0;
}
}
IFS1bits.IC7IF = 0;
}

```

### B.1.6 Init\_UART1

```

#include "p24fj128gb206.h"
extern int status_manual;
unsigned char buffer_u1[31]; //era 34
unsigned char OCC1R[5],OCC2R[5],OCC3R[5],OCC4R[5],OCC5R[5],OCC6R[5],OCC7R[5];
unsigned int n1 = 0;
//extern unsigned int status_ic_throtle;
void UART1Init()
{
UIBRG = 25; // 24Mhz osc, UIBRG = 12 = 57600 Baud // BRGH = 1 => 25 = 115200
UIMODE = 0x8008; //turn on module
UISTA = 0x8480; //set interrupts
IPC2bits.U1RXIP0 = 1;
IEC0bits.U1RXIE = 1; //Habilita interrupcao RX1
IFS0bits.U1RXIF = 0; //reset RX interrupt flag
}
void putcU1(char c)
{
if (!U1STAbits.UTXBF)
UITXREG = c;
while (!U1STAbits.TRMT);
}
//#1996$
void __attribute__((interrupt, no_auto_psv)) _U1RXInterrupt(void)
{
buffer_u1[n1] = UITXREG;
n1++;
IFS0bits.U1RXIF = 0; // clear RX interrupt flag
if (n1>1)
{
if ((buffer_u1[0]!=35) || (buffer_u1[n1-1]== ' ') || (n1>31) || buffer_u1[n1-1]==0x0A )
{
n1=0;
}
if ((buffer_u1[0]==35))

```

```
{
if ((n1>28) && (buffer_u1[29]==36))
{
n1 = 0;
if(status_manual == 1)
{
PORTCbits.RC14 = 1;
OCC6R[0] = (buffer_u1[21]);
OCC6R[1] = (buffer_u1[22]);
OCC6R[2] = (buffer_u1[23]);
OCC6R[3] = (buffer_u1[24]);
OC6R = atol(OCC6R);
OCC4R[0] = (buffer_u1[13]);
OCC4R[1] = (buffer_u1[14]);
OCC4R[2] = (buffer_u1[15]);
OCC4R[3] = (buffer_u1[16]);
OC4R = atol(OCC4R);
}

if(status_manual == 0)
{
PORTCbits.RC14 = 1;
OCC1R[0] = (buffer_u1[1]);
OCC1R[1] = (buffer_u1[2]);
OCC1R[2] = (buffer_u1[3]);
OCC1R[3] = (buffer_u1[4]);
OC1R = atol(OCC1R);
OCC2R[0] = (buffer_u1[5]);
OCC2R[1] = (buffer_u1[6]);
OCC2R[2] = (buffer_u1[7]);
OCC2R[3] = (buffer_u1[8]);
OC2R = atol(OCC2R);
OCC3R[0] = (buffer_u1[9]);
OCC3R[1] = (buffer_u1[10]);
OCC3R[2] = (buffer_u1[11]);
OCC3R[3] = (buffer_u1[12]);
OC3R = atol(OCC3R);
OCC4R[0] = (buffer_u1[13]);
OCC4R[1] = (buffer_u1[14]);
OCC4R[2] = (buffer_u1[15]);
OCC4R[3] = (buffer_u1[16]);
OC4R = atol(OCC4R);
OCC5R[0] = (buffer_u1[17]);
OCC5R[1] = (buffer_u1[18]);
OCC5R[2] = (buffer_u1[19]);
OCC5R[3] = (buffer_u1[20]);
OC5R = atol(OCC5R);
OCC6R[0] = (buffer_u1[21]);
OCC6R[1] = (buffer_u1[22]);
OCC6R[2] = (buffer_u1[23]);
OCC6R[3] = (buffer_u1[24]);
OC6R = atol(OCC6R);
OCC7R[0] = (buffer_u1[25]);
OCC7R[1] = (buffer_u1[26]);
```

```

OCC7R[2] = (buffer_u1[27]);
OCC7R[3] = (buffer_u1[28]);
OC7R = atol(OCC7R);
}
} // if((n1>28) && (buffer_u1[29]==36))
} // if((buffer_u1[0]==35))
} //if (n1>1)
} //void __attribute__((interrupt , no_auto_psv)) _U1RXInterrupt(void);

```

### B.1.7 TimerInterrrupts.c

```

#include "p24fj128gb206.h"

extern void putcU1();
extern unsigned int ULT1,ULT2,BAT1,BAT2,BAT3;
extern int Latitude_D,Longitude_D;
extern long int Latitude_m,Longitude_m,Time;
extern int ic1_period,ic2_period,ic3_period,ic4_period,ic5_period,ic6_period,ic7_period;

/*****          TIMER 3 Interrupt          *****/
void __attribute__((interrupt ,no_auto_psv)) _T3Interrupt( void )// 22ms
{
putcU1('\n');
putcU1('#');
putcU1(ULT1&0xFF);
putcU1((ULT1>>8)&0xFF);
putcU1(ULT2&0xFF);
putcU1((ULT2>>8)&0xFF);
putcU1(BAT1&0xFF);
putcU1((BAT1>>8)&0xFF);
putcU1(BAT2&0xFF);
putcU1((BAT2>>8)&0xFF);
putcU1(BAT3&0xFF);
putcU1((BAT3>>8)&0xFF);
putcU1(ic1_period&0xFF);
putcU1((ic1_period>>8)&0xFF);
putcU1(ic2_period&0xFF);
putcU1((ic2_period>>8)&0xFF);
putcU1(ic3_period&0xFF);
putcU1((ic3_period>>8)&0xFF);
putcU1(ic4_period&0xFF);
putcU1((ic4_period>>8)&0xFF);
putcU1(ic5_period&0xFF);
putcU1((ic5_period>>8)&0xFF);
putcU1(ic6_period&0xFF);
putcU1((ic6_period>>8)&0xFF);
putcU1(ic7_period&0xFF);
putcU1((ic7_period>>8)&0xFF);

IFS0bits.T3IF = 0;//Clear the Timer3 Interrupt Flag
}

```

## B.2 Códigos Diferenciados por Veículo

Nesta Dissertação foram utilizados dois helimodelos de tamanhos diferentes, os quais deveriam portar o estabilizador *Helicommand* e o *AuRoRa Board*. Devido ao espaço limitado dos VANTs, principalmente do T-REX 450, teve-se de alterar a ordem de conexão das entradas. Como o microcontrador utilizado possui a capacidade de remapear suas entradas e saídas, apenas a rotina responsável por este mapeamento precisou ser alterada. Desta forma, dependendo de qual veículo o circuito for adicionado, deve-se escolher um dos códigos listados abaixo. É possível também, adicionar este mesmo circuito em outros tipos de helicópteros, basta atentar-se às entradas e, se necessário, altera-las no código.

### B.2.1 Init\_IO\_450

```

/*****
/*                               IO Init                               */
*****/
#include "p24fj128gb206.h"

void IOInit_450(void)
{
    TRISEbits.TRISE3 = 0;    asm("nop");asm("nop");    //LED Yellow
    TRISCbits.TRISC14 = 0;   asm("nop");asm("nop");    //LED Blue
    TRISCbits.TRISC13 = 0;   asm("nop");asm("nop");    //LED Blue
    TRISBbits.TRISB14 = 0;

    __builtin_write_OSCCONL(OSCCON & 0xbf); // Unlock Registers

    /*****      Configure Output Functions      *****/
    RPOR13bits.RP27R      = 18;   asm("nop");asm("nop");    // RP27 pin#8
    RPOR9bits.RP19R       = 19;   asm("nop");asm("nop");    // RP29 pin#6
    RPOR13bits.RP26R      = 20;   asm("nop");asm("nop");    // RP26 pin#5
    RPOR10bits.RP21R      = 21;   asm("nop");asm("nop");    // RP21 pin#4
    RPOR10bits.RP20R      = 22;   asm("nop");asm("nop");    // RP20 pin#53
    RPOR12bits.RP25R      = 23;   asm("nop");asm("nop");    // RP25 pin#52
    RPOR11bits.RP22R      = 24;   asm("nop");asm("nop");    // RP22 pin#51
    RPOR3bits.RP7R        = 3;    asm("nop");asm("nop");    //TX1  pin#18
    RPOR4bits.RP9R        = 5;    asm("nop");asm("nop");    //TX2  pin#21
    RPOR14bits.RP29R      = 28;   asm("nop");asm("nop");    //TX3  pin#30

    /*****      Configure Input Functions      *****/
    RPINR7bits.IC1R       = 2;    asm("nop");asm("nop");    // RP23 pin#42
    RPINR7bits.IC2R       = 4;    asm("nop");asm("nop");    // RP24 pin#43
    RPINR8bits.IC3R       = 3;    asm("nop");asm("nop");    // RP11 pin#44
    RPINR8bits.IC4R       = 12;   asm("nop");asm("nop");    // RP12 pin#45
    RPINR9bits.IC5R       = 11;   asm("nop");asm("nop");    // RP3  pin#46
    RPINR9bits.IC6R       = 24;   asm("nop");asm("nop");    // RP4  pin#49
    RPINR10bits.IC7R      = 23;   asm("nop");asm("nop");    // RP2  pin#50
    RPINR17bits.U3RXR     = 14;   asm("nop");asm("nop");    // U3RX To RP14 pin#29

```

```

RPINR18bits.U1RXR      = 6;    asm("nop");asm("nop"); // U1RX To RP10 pin#17
RPINR19bits.U2RXR      = 8;    asm("nop");asm("nop"); // U2RX To RP9 pin#22

__builtin_write_OSCCONL(OSCCON | 0x40); // Lock Registers
}

```

## B.2.2 Init\_IO600

```

#include "p24fj128gb206.h"

void IOInit_600(void)
{
    TRISEbits.TRISE3 = 0;    asm("nop");asm("nop"); //LED Yellow
    TRISCbits.TRISC14 = 0;   asm("nop");asm("nop"); //LED Blue
    TRISCbits.TRISC13 = 0;   asm("nop");asm("nop"); //LED Blue
    TRISBbits.TRISB14 = 0;

    __builtin_write_OSCCONL(OSCCON & 0xbf); // Unlock Registers

    /***** Configure Output Functions *****/
    RPOR13bits.RP27R      = 18;    asm("nop");asm("nop"); // RP27 pin#8
    RPOR9bits.RP19R       = 19;    asm("nop");asm("nop"); // RP29 pin#6
    RPOR13bits.RP26R      = 20;    asm("nop");asm("nop"); // RP26 pin#5
    RPOR10bits.RP21R      = 21;    asm("nop");asm("nop"); // RP21 pin#4
    RPOR10bits.RP20R      = 22;    asm("nop");asm("nop"); // RP20 pin#53
    RPOR12bits.RP25R      = 23;    asm("nop");asm("nop"); // RP25 pin#52
    RPOR11bits.RP22R      = 24;    asm("nop");asm("nop"); // RP22 pin#51
    RPOR3bits.RP7R = 3;    asm("nop");asm("nop"); //TX1 pin#18
    RPOR4bits.RP9R = 5;    asm("nop");asm("nop"); //TX2 pin#21
    RPOR14bits.RP29R = 28;   asm("nop");asm("nop"); //TX3 pin#30

    /***** Configure Input Functions *****/
    RPINR7bits.IC1R = 23;    asm("nop");asm("nop"); // RP23 pin#50
    RPINR7bits.IC2R = 24;    asm("nop");asm("nop"); // RP24 pin#49
    RPINR8bits.IC3R = 11;    asm("nop");asm("nop"); // RP11 pin#46
    RPINR8bits.IC4R = 12;    asm("nop");asm("nop"); // RP12 pin#45
    RPINR9bits.IC5R = 3;    asm("nop");asm("nop"); // RP3 pin#44
    RPINR9bits.IC6R = 4;    asm("nop");asm("nop"); // RP4 pin#43
    RPINR10bits.IC7R = 2;    asm("nop");asm("nop"); // RP2 pin#42
    RPINR17bits.U3RXR = 14;   asm("nop");asm("nop"); // U3RX To RP14 pin#29
    RPINR18bits.U1RXR = 6;    asm("nop");asm("nop"); // U1RX To RP10 pin#17
    RPINR19bits.U2RXR = 8;    asm("nop");asm("nop"); // U2RX To RP9 pin#22

    __builtin_write_OSCCONL(OSCCON | 0x40); // Lock Registers
}

```

## B.3 Circuito da Placa AuRoRa Board

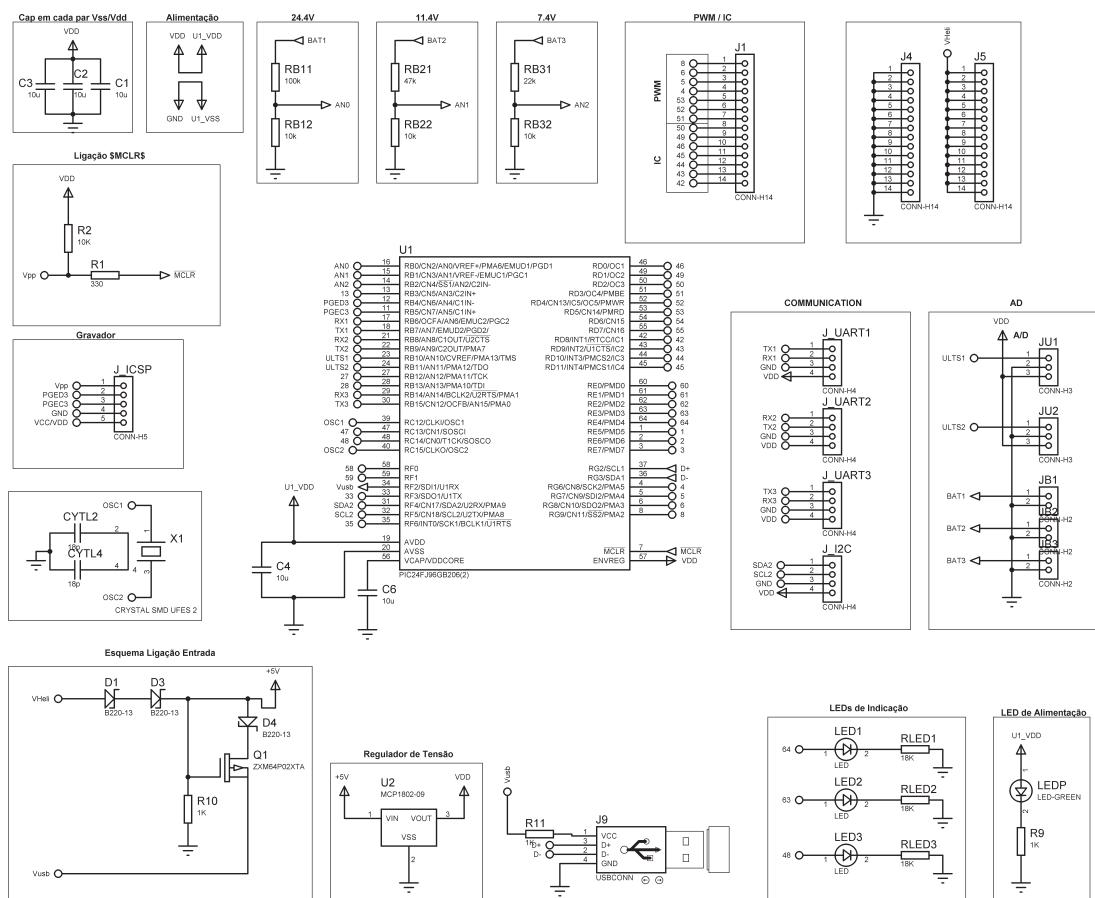


Figura 56: Diagrama do circuito da placa AuRoRa Board

## B.4 Camadas Físicas da Placa *AuRoRa Board*

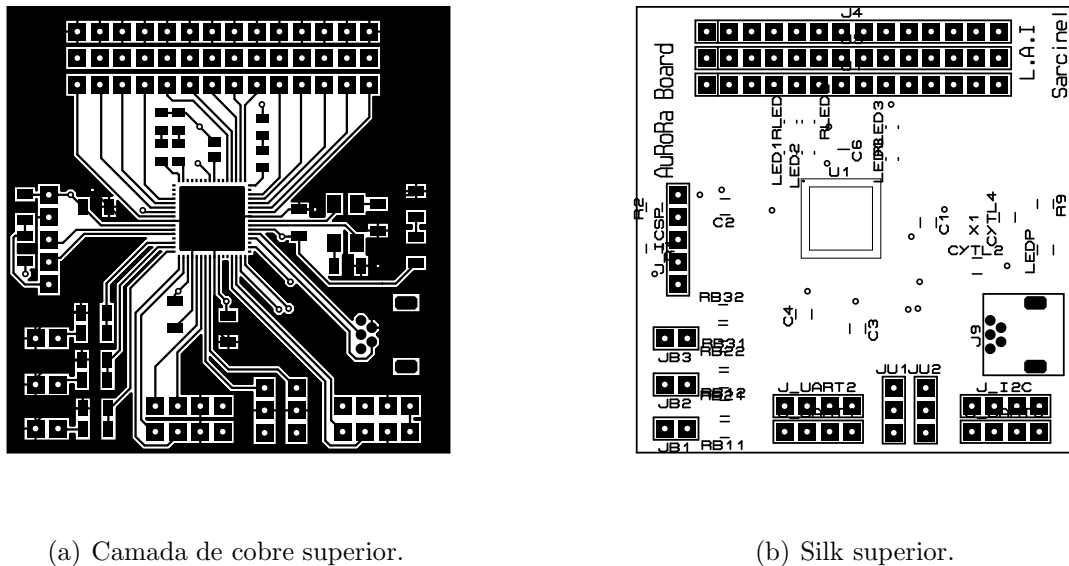


Figura 57: Parte superior da placa *AuRoRa Board*

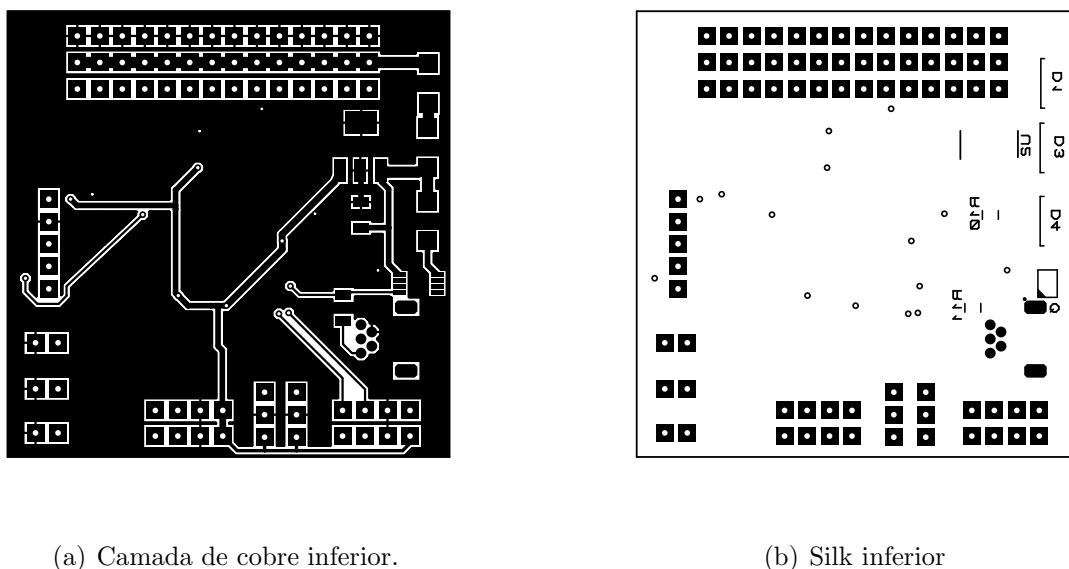


Figura 58: Parte inferior da placa *AuRoRa Board*



# *Referências*

- [1] BLYENBURGH, P. van. *UAS: The Global Perspective 2011/2012*. Paris, France: Blyenburgh & Co., 2011.
- [2] TJERNBERG, I.; LINDBERG, J.; HANSSON, K. *Cooperative Networked Control of Unmanned Air Vehicles*. Stockholm, Sweden, May 2011.
- [3] BOUABDALLAH, S.; MURRIERI, P.; SIEGWART, R. Design and control of an indoor micro quadrotor. In: *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA 2004)*. [S.l.: s.n.], 2004. v. 5, p. 4393–4398.
- [4] EISENBEISS, H. A mini unmanned aerial vehicle (uav): System overview and image acquisition. In: *Proceedings of the International Workshop on Processing and Visualization Using High-Resolution Imagery*. Pitsanulok, Thailand: [s.n.], 2004.
- [5] DIOS, J. M. de; OLLERO, A. An illumination-robust robot infrared vision system for robotics outdoor applications. In: *Proceedings of the International Symposium on Robotics and Applications (ISORA), World Automation Congress (WAC2004)*. Seville, Spain: [s.n.], 2004. v. 15, p. 413–418.
- [6] RATHINAM, S. et al. Vision based following of locally linear structures using an unmanned aerial vehicle. In: *Proceedings of the 44th IEEE Conference on Decision and Control, 2005 European Control Conference*. [S.l.: s.n.], 2005. p. 6085–6090.
- [7] BESTAOUI, Y.; SLIM, R. Maneuvers for a quad- rotor autonomous helicopter. In: *AIAA Conference and Exhibit*. Rohnert Park, California: [s.n.], 2007.
- [8] MACARTHUR, E.; MACARTHUR, D.; CRANE, C. Use of cooperative unmanned air and ground vehicles for detection and disposal of mines. In: *Proceedings of the VI Intelligent Systems in Design and Manufacturing*. [S.l.: s.n.], 2005. v. 5999, p. 94–101.
- [9] LINDEMUTH, M. et al. Sea robot-assisted inspection. *IEEE Robotics and Automation Magazine*, v. 18, n. 2, p. 96–107, June 2011.
- [10] TOURNIER, G.; VALENTIY, M.; HOWZ, J. Estimation and control of a quadrotor vehicle using monocular vision and moire patterns. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Keystone, Colorado: [s.n.], 2006.
- [11] KIM, S. K.; TILBURY, D. M. Mathematical modeling and experimental identification of a model helicopter. In: *Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit*. Boston, MA, USA: [s.n.], 1998. p. 203–213.
- [12] CASTILLO, P.; LOZANO, R.; DZUL, A. *Modelling and Control of Mini-Flying Machines*. USA: Springer, 2005.

- [13] TANG, Y.-R.; LI, Y. Design of an optimal flight control system with integral augmented compensator for a nonlinear uav helicopter. In: *Proceedings of the 10th World Congress on Intelligent Control and Automation (WCICA)*. [S.l.: s.n.], 2012. p. 3927–3932.
- [14] TISDALE, J. et al. The software architecture of the berkeley uav platform. In: *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*. [S.l.: s.n.], 2006. p. 1420–1425.
- [15] COELHO, T. et al. A fly-by-wireless uav platform based on a flexible and distributed system architecture. In: *Industrial Technology, 2006. ICIT 2006. IEEE International Conference on*. [S.l.: s.n.], 2006. p. 2359–2364.
- [16] RIBEIRO, L.; OLIVEIRA, N. M. F. Uav autopilot controllers test platform using matlab/simulink and x-plane. In: *Frontiers in Education Conference (FIE), 2010 IEEE*. [S.l.: s.n.], 2010. p. S2H-1–S2H-6. ISSN 0190-5848.
- [17] YOO, C. sun; KANG, Y. shin; PARK, B. jin. Hardware-in-the-loop simulation test for actuator control system of smart uav. In: *2010 IEEE International Conference on Control Automation and Systems (ICCAS)*. [S.l.: s.n.], 2010. p. 1729 –1732.
- [18] KANG, W. et al. Design and implementation of a hardware-in-the-loop simulation system for small-scale uav helicopters. In: *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*. [S.l.: s.n.], 2008. p. 29–34.
- [19] SALMAN, S.; PUTTIGE, V.; ANAVATTI, S. Real-time validation and comparison of fuzzy identification and state-space identification for a uav platform. In: *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*. [S.l.: s.n.], 2006. p. 2138–2143.
- [20] MUTTER, F. et al. Model-driven in-the-loop validation simulation-based testing of uav software using virtual environments. In: *Proceedings of the 18th IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS)*. [S.l.: s.n.], 2011. p. 269–275.
- [21] CAI, G. et al. Design and implementation of a hardware-in-the-loop simulation system for small-scale uav helicopters. In: *IEEE International Conference on Automation and Logistics (ICAL)*. [S.l.: s.n.], 2008. p. 29 –34.
- [22] ANTUNES, D.; SILVESTRE, C.; CUNHA, R. On the design of multi-rate tracking controllers: application to rotorcraft guidance and control. *International Journal of Robust and Nonlinear Control*, v. 20, p. 1879–1902, January 2010.
- [23] BRANDÃO, A. S.; SARCINELLI-FILHO, M.; CARELLI, R. A nonlinear underactuated controller for 3d-trajectory tracking with a miniature helicopter. In: *2010 IEEE International Conference on Industrial Technology (ICIT)*. [S.l.: s.n.], 2010. p. 1421–1426.

- [24] BRANDÃO, A. S.; SARCINELLI-FILHO, M.; CARELLI, R. High-level underactuated nonlinear control for rotorcraft machines. In: *Proceedings of the 2013 IEEE International Conference on Mechatronics - ICM2013*. Vicenza, Italy: [s.n.], 2013. p. 279–285.
- [25] ZEIN-SABATTO, S.; ZHENG, Y. Intelligent flight controllers for helicopter control. In: *Proceedings of the International Conference on Neural Networks*. [S.l.: s.n.], 1997. v. 2, p. 617–621.
- [26] SIRA-RAMIREZ, H.; CASTRO-LINARES, R.; LICEAGA-CASTRO, E. A liouvilian systems approach for the trajectory planning-based control of helicopter models. v. 10, n. 4, p. 301–320, 2000.
- [27] PALOMINO, A. et al. Control strategy using vision for the stabilization of an experimental pvtol aircraft setup. In: *Proceedings of the 42nd IEEE Conference on Decision and Control*. [S.l.: s.n.], 2003.
- [28] SANTOS, W. V. dos. *Modelagem, identificação e controle de altitude de um helicóptero em escala reduzida*. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, Rio de Janeiro - RJ, Brasil, Abril 2005.
- [29] BUDIYONO, A.; WIBOWO, S. S. Optimal tracking controller design for a small scale helicopter. *Journal of Bionic Engineering*, v. 4, p. 271–280, 2007.
- [30] KAHN, A. D.; FOCH, R. J. Attitude command attitude hold and stability augmentation system for a small-scale helicopter uav. In: *Proceedings of the 22nd Digital Avionics Systems Conference*. [S.l.: s.n.], 2003.
- [31] BUSKEY, G. et al. Helicopter automation using a low-cost sensing system. In: *Proceedings of the Australasian Conference on Robotics and Automation*. Brisbane, Australia: [s.n.], 2003.
- [32] MARCONI, L.; NALDI, R. Robust nonlinear control of a miniature helicopter for aerobatic maneuvers. In: *Proceedings of the 32nd Rotorcraft Forum*. [S.l.: s.n.], 2006.
- [33] MARTINI, A.; LÉONARD, F.; ABBA, G. Dynamic modelling and stability analysis of model-scale helicopters under wind gust. *Journal of Intelligent and Robotic Systems*, Kluwer Academic Publishers, Hingham, MA, USA, v. 54, n. 4, p. 647–686, 2009. ISSN 0921-0296.
- [34] BECKMANN, E. D.; BORGES, G. A. Nonlinear modeling, identification and control for a simulated miniature helicopter. In: *Proceedings of the Latin American Robotic Symposium*. Los Alamitos, CA, USA: IEEE Computer Society, 2008. p. 53–58.
- [35] BRANDÃO, A. S. et al. 3-d path-following with a miniature helicopter using a high-level nonlinear underactuated controller. In: *Proceedings of the 9th IEEE International Conference on Control and Automation - ICCA'11*. Santiago, Chile: [s.n.], 2011. p. 434–439.
- [36] BRANDÃO, A. S. et al. Modelagem e controle não linear subatuado de um quadrotor: Parte 1. In: *Anais do XIX Congresso Brasileiro de Automática - CBA2012*. Campina Grande, PB: SBA, 2012. p. 449–454.

- [37] PETTERSEN, R.; MUSTAFIC, E.; FOGH, M. *Nonlinear Control Approach to Helicopter Autonomy*. Dissertação (Mestrado) — Institute of Electronic Systems, Department of Control Engineering of the Aalborg University, 2005.
- [38] HASHIMOTO, S. et al. Robust control design based on identified model for autonomous flight system of an unmanned helicopter. In: *Proceedings of the 27th Annual Conference of the IEEE Industrial Electronics Society - IECON'01*. Denver, CO, USA: [s.n.], 2001.
- [39] KOO, T. J.; SASTRY, S. Output tracking control design of a helicopter model based on approximate linearization. In: *Proceedings of the 37th Conference on Decision and Control*. Tampa, Florida USA: [s.n.], 1998. p. 3635–3640.
- [40] CHRIETTE, A.; HAMEL, T.; MAHONY, R. Visual servoing for a scale model autonomous helicopter. In: *Proceedings of the IEEE International conference on robotics and Automation*. Seul, Korea: [s.n.], 2001. v. 2, p. 1701–1706.
- [41] KONDAK, K. et al. Mechanical model and control of an autonomous small size helicopter with a stiff main rotor. In: *Proceedings of the International Conference on Intelligent Robots and Systems*. Sendai, Japan: [s.n.], 2004.
- [42] OH, S.-R. et al. Autonomous helicopter landing on a moving platform using a tether. In: *Proceedings of the IEEE International Conference on Robotics and Automation - ICRA2005*. Barcelona, Spain: [s.n.], 2005.
- [43] RAFFO, G. V.; ORTEGA, M. G.; RUBIO, F. R. An integral predictive/nonlinear  $\mathcal{H}_\infty$  control structure for a quadrotor helicopter. *Automatica*, v. 46, p. 29–39, 2010.
- [44] AHMED, B.; POTA, H. R.; GARRATT, M. Flight control of a rotary wing uav using backstepping. *International Journal of Robust and Nonlinear Control*, v. 20, p. 639–658, January 2010.
- [45] KONDAK, K. et al. Autonomously flying vtol-robots: Modeling and control. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Rome, Italy: [s.n.], 2007. p. 736–741.
- [46] SANTANA, L. V. et al. Hovering control of a miniature helicopter attached to a platform. In: *Proceedings of the 20th IEEE International Symposium on Industrial Electronics - ISIE2011*. Gdansk, Poland: [s.n.], 2011. p. 2231–2236.
- [47] BRANDÃO, A. *Projeto de Controladores Não Lineares para Voo Autônomo de Veículos Aéreos de Pás Rotativas*. Tese (Doutorado) — Universidade Federal do Espírito Santo, Vitória, ES, Abril 2013.
- [48] MARTINS, A. S. *Instrumentação e controle de altitude para helimodelo montado em uma plataforma de testes*. Dissertação (Mestrado) — Faculdade de Tecnologia, Departamento de Engenharia Elétrica, Universidade de Brasília, 2008.
- [49] SONG, B. et al. Nonlinear dynamic modeling and control of a small-scale helicopter. *International Journal of Control, Automation, and Systems*, v. 8, n. 3, p. 534–543, 2010.

- [50] BERNARD, M. et al. Elaborated modeling and control for an autonomous quad-rotor. In: *21st Bristol UAV Systems Conference*. [S.l.: s.n.], 2007.
- [51] GAVRILETS, V.; METTLER, B.; FERON, E. *Dynamic Model for a Miniature Aerobatic Helicopter*. [S.l.], 2004.
- [52] CAI, G. et al. Design and implementation of a robust and nonlinear flight control system for a unmanned helicopter. *Mechatronics*, v. 21, p. 803–820, 2011.
- [53] BÉJAR, M.; OLLERO, A.; CUESTA, F. Modeling and control of autonomous helicopters. In: BONIVENTO, C. et al. (Ed.). *Advances in Control Theory and Applications*. [S.l.]: Springer Berlin Heidelberg, 2007, (Lecture Notes in Control and Information Sciences, v. 353). p. 1–29.
- [54] VILCHIS, J. C. A. et al. Nonlinear modelling and control of helicopters. *Automatica*, v. 39, p. 1583–1596, 2003.
- [55] SPARKFUN. *9DOFIMU*, Acesso em: 15 Abril. 2013. Disponível em: <<https://www.sparkfun.com/products/10736>>.
- [56] SPARKFUN. *BMP085*, Acesso em: 15 Abril. 2013. Disponível em: <<https://www.sparkfun.com/products/11282>>.
- [57] SPARKFUN. *EZ4 MaxBotix*, Acesso em: 15 Abril. 2013. Disponível em: <<https://www.sparkfun.com/products/9495>>.
- [58] MICROCHIP. *PIC24FJ256GB206*, Acesso em: 15 Abril. 2013. Disponível em: <<http://ww1.microchip.com/downloads/en/devicedoc/39897b.pdf>>.
- [59] GOKTOGAN, A. et al. Real time multi-uav simulator. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2003. v. 2, p. 2720 – 2726 vol.2. ISSN 1050-4729.
- [60] BONIVENTO, C. et al. Rapid prototyping of automated manufacturing systems by software-in-the-loop simulation. In: *Chinese Control and Decision Conference (CCDC)*. [S.l.: s.n.], 2011. p. 3968 –3973.
- [61] DEMERS, S.; GOPALAKRISHNAN, P.; KANT, L. A generic solution to software-in-the-loop. In: *Proceedings of the IEEE Military Communications Conference (MILCOM)*. [S.l.: s.n.], 2007. p. 1 –6.
- [62] FRYE, M.; BHANDARI, S.; COLGREN, R. The raptor 50 6-dof simulation environment for flight control research. In: *Proceedings of the American Control Conference, 2006*. [S.l.: s.n.], 2006. p. 6 pp.
- [63] BRANDÃO, A. S. et al. Decentralized control of a formation involving a miniature helicopter and a team of ground robots based on artificial vision. In: *Proceedings of the 2010 Latin American Robotics Symposium and Intelligent Robotics Meeting - LARS2010*. São Bernardo do Campo/SP, Brasil: IEEE, 2010. p. 126–131.
- [64] PIZETTA, I. H. B. et al. High-level flight controllers applied to helicopter navigation: A comparative study. In: *Proceedings of the 2012 Latin American Robotics Symposium - LARS2012*. Fortaleza,CE: [s.n.], 2012. p. 162–167.

- [65] PIZETTA, I. et al. Controladores de Vuelo de Alto Nivel Aplicados a la Navegación de Helicópteros : Un abordaje comparativo. In: *Anales de la XIV Reunión de Trabajo en Procesamiento de la Información y Control (RPIC 2011)*. Oro Verde, Entre Ríos, Argentina: [s.n.], 2011. p. 584–589.
- [66] BRANDAO, A. S. et al. Comparação entre Controladores de Voo de Alto Nível Aplicados à Navegação Autônoma de um Helicóptero Miniatura. In: *Anais do X Simpósio Brasileiro de Automação Inteligente - SBAI2011*. São João del Rei, MG: [s.n.], 2011. p. 1316–1321.
- [67] BRANDÃO, A. S. et al. A strategy to avoid dynamic and static obstacles for robotic wheelchairs. In: *Proceedings of the IEEE International Symposium on Industrial Electronics*. Bari, Italy: [s.n.], 2010. p. 3553–3558.
- [68] BRANDÃO, A.; SARCINELLI-FILHO, M.; CARELLI, R. Um controlador de voo 3d não linear baseado na dinâmica inversa do modelo sub-atuado de um helicóptero miniatura. In: *Anais do XVIII Congresso Brasileiro de Automática - CBA2010*. Bonito, MS: [s.n.], 2010. p. 1852–1859.
- [69] KIM, K.-H.; CHO, H. S. An obstacle avoidance method for mobile robots based on fuzzy decision-making. *Robotica*, v. 24, n. 5, p. 567–578, 2006.
- [70] ROY, T. et al. Robust altitude control for a small helicopter by considering the ground effect compensation. In: *Intelligent Control and Automation (WCICA), 2012 10th World Congress on*. [S.l.: s.n.], 2012. p. 1796–1800.
- [71] WANG, X.; GUO, J.; CUI, N. Adaptive extended kalman filtering applied to low-cost mems imu/gps integration for uav. In: *Mechatronics and Automation, 2009. ICMA 2009. International Conference on*. [S.l.: s.n.], Aug. p. 2214–2218.
- [72] RIGATOS, G. Technical analysis and implementation cost assessment of sigma-point kalman filtering and particle filtering in autonomous navigation systems. In: *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*. [S.l.: s.n.], May. p. 1–5. ISSN 1550-2252.
- [73] OH, S.-M. Multisensor fusion for autonomous uav navigation based on the unscented kalman filter with sequential measurement updates. In: *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2010 IEEE Conference on*. [S.l.: s.n.], Sept. p. 217–222.
- [74] YUN, X. et al. Implementation and experimental results of a quaternion-based kalman filter for human body motion tracking. In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. [S.l.: s.n.], April. p. 317–322.
- [75] MARINS, J. et al. An extended kalman filter for quaternion-based orientation estimation using marg sensors. In: *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)*. [S.l.: s.n.], 2001. v. 4, p. 2003–2011.

- 
- [76] SANTANA, L. V. et al. Uma estrutura sensorial e de comunicação para o controle de altitude e guinada de um mini-helicóptero autônomo. In: *Anais do XVIII Congresso Brasileiro de Automática - CBA2010*. Bonito - MS, Brasil: [s.n.], 2010. p. 1840–1845.
- [77] KUIPERS, J. B. Quaternions and rotations sequences: A primer with applications to orbits, aerospace and virtual reality. In: *Princeton University Press*. [S.l.: s.n.], 1999.
- [78] COOKE, J. M. e. a. Flight simulation dynamic modeling using quaternions. *Presence*, v. 1, p. 404–4020, 1992.
- [79] CARUSO, M. J. Applications of magnetic sensors for low cost compass systems. In: *[S.l.]*. [S.l.: s.n.], 2002.
- [80] MADGWICK, S. O. H.; VAIDYANATHAN, R.; HARRISON, A. *An Efficient Orientation Filter for IMU and MARG Sensor Arrays*. Department of Mechanical Engineering, University of Bristol: [s.n.], 2010.
- [81] GEBRE-EGZIABHER, D.; HAYWARD, R. C.; POWELL, J. D. Design of multi-sensor attitude determination systems. *Aerospace and Electronic Systems Society*, (40):2, p. 627–649, 2004.
- [82] FOXLIN, E. Inertial head-tracker sensor fusion by a complementary separate-bias kalman filter. In: *Proceedings of the Virtual Reality Annual International Symposium the IEEE 1996*. [S.l.: s.n.], 1996. p. 185–194,267.
- [83] MAHONY, R.; HAMEL, T.; J.-M.PIMLIN. Nonlinear complementary filters on the special orthogonal group. *IEEE transactions on Automatic Control*, v. 53, p. 1203–1218, 2008.